Wing IDE 101 Benutzerhandbuch
Wing IDE 101

Wingware
www.wingware.com

Version 2.1.3
October 12, 2006

Inhalt

Einleitung

- 1.1. Unterstützte Plattformen
- 1.2. Unterstützte Python-Versionen
- 1.3. Grundvoraussetzungen für die Installation
- 1.4. Installation
- 1.5. Ausführung des IDEs
- 1.6. Verzeichnis der Benutzereinstellungen
- 1.7. Aufrüsten (Upgrade)
  - 1.7.1. Ein gescheitertes Upgrade beheben
- 1.8. Erweiterte Installation
  - 1.8.1. Installation zusätzlicher Dokumentation
  - 1.8.2. Installationshinweise für Linux
- 1.9. Wing IDE entfernen
- 1.10. Fehlerbehebung
  - 1.10.1. Fehlerbehebung für Startfehler
  - 1.10.2. Probleme in Microsoft Windows
  - 1.10.3. Wing IDE beschleunigen
  - 1.10.4. Fehlerbehebung öffnungs-Fehler der Dateinamen mit Leerzeichen

Anpassung

- 2.1. Optionen der Benutzeroberfläche
  - 2.1.1. Layout der Benutzeroberfläche
  - 2.1.2. Änderung der Textanzeige
  - 2.1.3. Einstellung des insgesamten Anzeigethemas
- 2.2. Einstellungen
- 2.3. Editor-Individualitäten

Source-Code-Editor

- 3.1. Syntax-Farbmarkierung
- 3.2. Rechtsklick-Menü des Editors
- 3.3. Source-Code-Navigation
- 3.4. Dateistatus und nur lesbare Dateien
- 3.5. Vorübergehende vs. nicht vorübergehende Editoren

3.6. Klammernsuche
3.6.1. Automatisch Einrücken
3.6.2. Die Tab-Taste
3.6.3. Blockeinrückung ändern
3.7. Auto-Vervollständigung
3.8. Hinweise zu Kopieren/Einfügen
3.9. Suchen/Ersetzen
3.9.1. Schnellsuche mit der Werkzeugleiste
3.9.2. Search Tool
3.10. Source-Code-Analyse
3.10.1. Analyse-Cache
Debugger
4.1. Haltepunkte setzen
4.2. Debuggen starten
4.3. Debugger-Status
4.4. Ablaufsteuerung
4.5. Stack anzeigen
4.6. Debug-Daten anzeigen
4.6.1. Ansicht der Stack-Daten
4.6.1.1. Optionen des Popup-Menüs
4.6.2. Probleme bei der Behandlung von Werten
4.7. Interaktive Python-Shell
4.8. Debug-Prozess-I/O
Lizenzinformationen
5.1. Wing IDE Software-Lizenz
5.2. Open Source Lizenzinformationen

```
Wingware
P.O. Box 1937
Brookline, MA 02446
United States of America
```

Einleitung

Vielen Dank, dass Sie sich für Wing IDE 101 von Wingware entschieden haben! Das Handbuch wird Ihnen beim Starten helfen und dient als Referenz für das gesamte Funktionsset des Produkts.

Das Handbuch ist nach Hauptfunktionsbereichen von Wing IDE gegliedert, was den Projektmanager, Source-Code-Editor, und Debugger beinhaltet. Mehrere Anhänge dokumentieren das gesamte Befehlsset, stellen Hinweise zu Ressourcen und Tipps für Wing- und Python-Nutzer bereit und führen die volle Software-Lizenz auf.

Der Rest dieses Kapitels beschreibt, wie Sie Wing IDE 101 installieren und starten. Wenn Sie Handbücher nicht gern lesen, sollten Sie in der Lage sein, das Produkt zum Laufen zu bringen, indem Sie nur dieses Kapitel lesen.

Schlüsselkonzepte
In dem Handbuch sind Schlüsselkonzepte, wichtige Hinweise und nicht offensichtliche Funktionen genauso wie dieser Paragraph hervorgehoben. Wenn Sie den Text nur überfliegen, dann suchen Sie nach diesen Markierungen.

Beachten Sie, dass der gesamte Inhalt des Handbuchs auch innerhalb von Wing IDE durch den Hilfemanager verfügbar ist.

1.1. Unterstützte Plattformen

Diese Version von Wing IDE ist für Microsoft Windows, Linux und Mac OS X erhältlich. Außerdem steht es für einige andere Betriebssysteme zur Verfügung, für die **‘Builds von anderen Nutzern bereitgestellt‘__** wurden oder bei denen Kunden bereit sind, das Produkt vom Source-Code zu kompilieren.

Microsoft Windows

Wing IDE unterstützt Windows 98 mit IE5+ (*), ME mit IE5+ (*), NT4 mit IE5+ (*), 2K, XP sowie 2003 Server. Windows 95 wird nicht unterstützt.

(*) In Windows 98 können ME und NT4 Installationen mit **'diesem Patch von Microsoft'__** höchstwahrscheinlich verwendet werden, anstatt zu IE5 aufzurüsten.

Linux/Intel

Wing IDE läuft auf Linux-Versionen mit glibc2.2 oder höher (alles, das ungefähr 3 Jahre alt oder neuer ist sollte funktionieren; zum Beispiel RedHat 7.1+, Mandrake 8.0+, SuSe 7.1+ und Debian 3.0+).

In Suse müssen Sie die gmp- und python-Pakete installieren oder Python vom Source-Code installieren, da Python hier standardmäßig nicht installiert ist.

In Debian können Sie das Wing IDE RPM-Paket in ein Debian-freundliches Paket umwandeln, indem Sie das `alien` Modul verwenden. Installieren Sie das `alien`-Paket und führen dann `alien -d wingide-*.i386.rpm` aus, gefolgt von `dpkg -i wingide-*.deb`. Alternativ können Sie den Wing IDE tar-Datei-Installierer verwenden.

Mac OS X

Wing IDE läuft auf Mac OS X 10.1+. Wing IDE für OS X erfordert außerdem einen X11 Server und Fenstermanager. Siehe **'OS X Schnellstart-Anleitung'__** für Einzelheiten.

Für Mac OS X wird nur Python 2.2 oder höher unterstützt. Version 10.3 oder höher von OS X werden jedoch mit einer bereits installierten Standardversion von Python geliefert.

Andere Plattformen

Kunden können Wing IDE vom Source-Code kompilieren, wenn Sie es auf anderen Betriebssystemen (wie Linux PPC, Free BSD oder Solaris) verwenden möchten. Dies erfordert die Unterzeichnung einer **'Geheimhaltungsvereinbarung'__**.

Einige **'Builds für Wing IDE'__**, die von anderen Nutzern bereitgestellt wurden, sind auch für andere Betriebssysteme verfügbar.

1.2. Unterstützte Python-Versionen

Vor der Installation von Wing, müssen Sie zunächst **'Python 1.5.2'__**, **'Python 2.0'__**, **'Python 2.1'__**, **'Python 2.2'__**, **'Python 2.3'__** oder **'Python 2.4'__** herunterladen und installieren, wenn Sie nicht bereits eine dieser Versionen auf Ihrer Maschine haben.

In Windows muss Python mit einem der Installer von python.org installiert werden (oder es muss vom Source-Code erstellt werden, wenn gewünscht).

In Linux kommen die meisten Distributionen mit Python. Die Installation von Py-

thon ist normalerweise nur in Suse, RedHat 6.0 oder einer speziell angepassten Linux-Installation erforderlich.

In Suse Linux können Sie die gmp- und Python-Pakete installieren, die mit Ihrer Distribution kommen oder Sie installieren Material, das von den oben genannten Links bereitgestellt wird.

In RedHat 6.0 müssen Sie Python 1.5.2 oder höher installieren und dies zur Ausführung Ihres Debug-Programms und dem tar-Datei-Installierer (wenn nicht von RPM installiert) verwenden. Wing funktioniert mit der standardmäßigen 1.5.1 Installation, die mit RedHat 6.0 kommt, nicht.

Auf Mac OS X unterstützt Wing IDE nur Python 2.2 oder höher.

1.3. Grundvoraussetzungen für die Installation

Zur Ausführung von Wing IDE müssen Sie die folgenden Dinge erwerben und installieren, wenn Sie nicht bereits auf Ihrem System vorhanden sind:

Grundvoraussetzungen für alle Plattformen:

- Version von Wing IDE, entweder **'heruntergeladen'__** oder von CD
- Eine **'unterstützte Python-Version'__**
- Eine funktionierende TCP/IP Netzwerk-Konfiguration
- Internet Explorer, Netscape oder anderer Web-Browser (optional)
- **'Adobe Acrobat Reader 4.0.5'__** oder höher (optional)

Zusätzliche Grundvoraussetzungen für Mac OS X:

- Ein X Window-Server, wie **'Apple X11 für OS X'__** oder **'XDarwin'__**
- Ein Fenstermanager. Apple's Server beinhaltet einen; andere Optionen sind **'Window Maker'__** und **'OroborOSX'__**

1.4. Installation

Versichern Sie sich vor der Installation von Wing IDE, dass die **'notwendigen Grundvoraussetzungen'__** installiert sind. Wenn Sie eine vorherige Version aufrüsten, lesen

Sie zuerst den Abschnitt **‘Aufrüsten‘__**. Eine schnelle Einleitung zu den Funktionen von Wing IDE ist in der **‘Wing IDE Schnellstart-Anleitung‘__** zu finden.

Hinweis: Auf allen Plattformen wird der Installationsort von Wing IDE als `WINGHOME` bezeichnet.

Windows 98se, NT 4, Windows 2000 und Windows XP

Installieren Sie Wing IDE, indem Sie die heruntergeladene Executable ausführen. Wing's Dateien werden standardmäßig in `C:\Programme\Wing IDE` installiert, aber dieser Speicherort kann während der Installation verändert werden. Wing wird außerdem das **‘Verzeichnis der Benutzereinstellungen‘__** an der für Ihre Windows-Version entsprechenden Stelle anlegen. Es wird dazu verwendet, Einstellungen und andere Einrichtungen zu speichern.

Linux

Um das RPM zu installieren, müssen Sie zu Root wechseln und `rpm -i wingide-101-2.1.3-1.i386.rpm` eintippen. Das IDE wird in `/usr/lib/wingide2.1` installiert mit Executables in `/usr/bin`. Die Wing IDE Executable ist `/usr/bin/wing-1012.1`.

Um die tar-Datei zu installieren, müssen Sie `tar -zxvf wingide-101-2.1.3-1-i386-linux.tar.gz` eintippen, um aus dem tar-Archiv zu extrahieren. Gehen Sie dann mit `cd` zum neu erstellten Verzeichnis `wingide-101-2.1.3-1-i386-linux`, tippen `./wing-install.py` ein und beantworten die Fragen, um zu bestimmen, wo die Programmdateien gespeichert werden sollen.

Die Wing IDE Executable heißt `wing-1012.1`. Sie werden Ihre PATH-Umgebung ändern müssen, wenn Sie die Executable in einem Verzeichnis gespeichert haben, das nicht bereits in Ihrem Pfad ist. Sie werden auch eine neue Shell öffnen müssen oder anfordern, dass Ihre Shell die Festplatte nach Executables neu durchsucht. (zum Beispiel mit `rehash` in tsch).

Wing wird das **‘Verzeichnis der Benutzereinstellungen‘__** in `~/.wing101-2` anlegen. Es wird verwendet, um Einstellungen und andere Einrichtungen zu speichern.

Weitere Informationen finden Sie in **‘Linux Installationsdetails‘__**

Mac OS X

Auf Mac OS X erfordert Wing IDE, dass Sie zuerst (a) Python 2.2 oder höher (frühere Versionen werden nicht funktionieren) und (b) einen X Server und Fenstermanager installieren. Einzelheiten zur Installation und Ausführung auf OS X finden Sie unter **‘Wing IDE für OS X‘__**.

1.5. Ausführung des IDEs

Für eine schnelle Einführung zu Wing's Funktionen lesen Sie bitte die **'Wing IDE Schnellstart-Anleitung'__**. Für einen sanfteren, ausführlicheren Start lesen Sie bitte das **'Wing IDE Tutorial'__**.

In Windows starten Sie Wing IDE aus der Programmgruppe des Startmenüs. Sie können Wing auch von der Befehlszeile mit `wing-101` (in `WINGHOME` gelegen) starten.

In Linux/Unix führen Sie einfach `wing-1012.1` aus (in `WINGHOME` gelegen).

In Mac OS X starten Sie zuerst Ihren X Windows-Server und Fenstermanager. Wenn das erledigt ist, starten Sie Wing IDE mit einem Doppelklick auf den Anwendungsordner. Wenn Sie Wing von der Befehlszeile duch Verwendung von `Contents/MacOS/wing` im Wing IDE Anwendungsordner starten, dann müssen sie Ihre `DISPLAY` Umgebungsvariable setzen.

1.6. Verzeichnis der Benutzereinstellungen

Wing wird bei der ersten Ausführung automatisch Ihr `Verzeichnis der Benutzereinstellungen` anlegen. Dieses Verzeichnis speichert Ihre Lizenz, Ihre Einstellungen, automatisch gespeicherte Dateien, den Source-Analyse-Cache, zuletzt verwendete Listen und andere, von Wing intern verwendete Dateien. Wenn dieses Verzeichnis nicht erstellt werden kann, wird Wing beenden.

Das Einstellungsverzeichnis wird an einem Ort erstellt, der für Ihr Betriebssystem geeignet ist. Der Speicherort ist in der **'Über Wing IDE Box'__**, die über das Menü `Hilfe` erreicht werden kann, als Ihr `Einstellungsverzeichnis` gelistet.

Dies sind die von Wing verwendeten Speicherorte:

- **Linux/Unix** -- `~/.wing101-2` (ein Unterverzeichnis Ihres Home-Verzeichnisses)
- **Windows** -- In `Wing IDE 2` innerhalb des `Application Data` Ordners. Der Speicherort variiert in Abhängigkeit von der Windows-Version. Die unten aufgelisteten Verzeichnisse sind die Voreinstellungen für die englischsprachige Version von Windows und ein Systemlaufwerk `c:`. Sehen Sie in den Informationen nach, die in der Dialogbox `Über Wing IDE` aufgelistet sind, um das tatsächlich verwendete Verzeichnis zu bestimmen.

  Windows 98 und ME -- c:\Windows\Application Data

  Windows NT -- c:\WINNT\Profiles\${Benutzername}\Application Data

Windows 2000 und XP -- c:\Documents and Settings\${Benutzername}\Application Data

1.7. Aufrüsten (Upgrade)

Wenn Sie Wing aufrüsten und vorher Patch-Dateien installiert haben, lesen Sie diese **'zusätzlichen Informationen'__**, um Probleme während des Upgrades zu vermeiden.

Wenn Sie innerhalb der gleichen Unterversionsnummer von Wing IDE aufrüsten (zum Beispiel von 1.1.8 auf 1.1.10), wird dies Ihre vorherige Installation ersetzen. Sobald Sie aufgerüstet haben, sollten Ihre vorherigen Einstellungen noch vorhanden sein und Sie sollten in der Lage sein, Wing sofort zu starten.

Wenn Sie über Haupt-Releases aufrüsten (zum Beispiel von 2.0 auf 2.1), wird dies eine neue Version installieren, die neben der alten Version existiert.

Beachten Sie, dass die Einstellungen in Wing IDE 2.x vollkommen separat von den Werten sind, die Sie in allen früheren Wing IDE 1.1 Installationen bestimmt haben. Wing 2.x wird die anfänglichen Werte auf allen in Ihrer 1.1 Installation gefundenen Werten basieren, aber dies wird nur das erste Mal gemacht, wenn Sie Wing IDE 2.x starten.

Um ein Upgrade zu installieren, folgen Sie den Schritten, die im Abschnitt **'Installation'__** beschrieben sind.

1.7.1. Ein gescheitertes Upgrade beheben

In seltenen Fällen, wenn Sie innerhalb von Unterversionen aufrüsten (zum Beispiel von 2.0 zu 2.0.1), kann das Aufrüsten daran scheitern, alte Dateien zu überschreiben, was zu zufälligen oder bizarren Verhalten oder Abstürzen führt. Sie beheben dieses Problem, indem Sie Wing vollständig deinstallieren und verbleibende Dateien manuell entfernen, bevor Sie das Upgrade noch einmal installieren.

Windows

Für die Deinstallation in Window verwenden Sie die Schaltfläche `Programme Hinzufügen/Entfernen`, um Wing IDE zu deinstallieren. Gehen Sie dann in das Verzeichnis, in welchem Wing platziert war und entfernen manuell alle verbleibenden Ordner und Dateien.

Linux RPM

Wenn Sie Wing IDE für Linux von RPM installiert haben, erteilen Sie den Befehl `rpm -e wingide`. Gehen Sie dann in `/usr/lib/wingide` und entfernen manuell alle verbleibenden Dateien und Verzeichnisse.

Linux Tar

Wenn Sie Wing IDE für Linux aus der tar-Distribution installiert haben, suchen Sie Ihr Wing Installationsverzeichnis und führen Sie das dort platzierte Skript `wing-uninstall` aus. Ist das erledigt, entfernen Sie manuell alle verbleibenden Dateien und Verzeichnisse.

Mac OS X

Auf Mac OS X entpacken Sie einfach ein Archiv, um die Installation zu bilden. Probleme können auftreten, wenn Sie dies über eine bestehende Installation machen. Um das zu vermeiden, entpacken Sie an einer anderen Stelle.

1.8. Erweiterte Installation

Dieser Abschnitt beschreibt Installationsoptionen für den fortgeschrittenen Nutzer.

1.8.1. Installation zusätzlicher Dokumentation

Wenn Sie Linux/Unix verwenden, ist das Python-Handbuch in den meisten Installationen nicht enthalten. Sie können allerdings lokale Kopien dieser Seiten herunterladen und installieren.

Platzieren Sie die höchste Ebene des **'HTML-formatierten Python-Handbuches'__** (dort, wo `index.html` zu finden ist) in `python-manual/#.#` in Ihrer Wing IDE Installation. Ersetzen Sie `#.#` mit der Haupt- und Unterversion des entsprechenden Python-Interpreters. (Verwenden Sie beispielsweise für das Python 2.3.x Handbuch `python-manual/2.3`.)

Wenn dies vorgenommen wurde, wird Wing die Kopie auf dem lokalen Laufwerk nutzen, anstatt ins Internet zu gehen, wenn das Python-Handbuch aus dem Hilfemenü ausgewählt wird.

1.8.2. Installationshinweise für Linux

In Linux kann Wing von RPM oder vom tar-Archiv installiert werden. Verwenden Sie die letztere Variante, wenn Sie auf Ihrer Machine keinen Root-Zugang haben oder Wing irgendwo anders als `/usr/lib/wingide-101` installieren möchten.

Installation von RPM:

Auf RPM-basierten Systemen, wie RedHat und Mandrake, kann Wing von einem RPM-Paket installiert werden. Um es zu installieren, führen Sie `rpm -i wingide-101-2.1.3-1.i386.rpm` als Root aus oder verwenden Sie Ihr Lieblingsverwaltungstool für RPM, um das RPM zu installieren. Die meisten Dateien für Wing werden im Verzeichnis `/usr/lib/wingide-101` gespeichert und ein Link für den `wing-1012.1` Befehl ist im Verzeichnis `/usr/bin` platziert.

Der Installationsort von Wing wird als `WINGHOME` bezeichnet. Wenn Sie von RPM installiert haben, wird es immer `/usr/lib/wingide-101` sein.

Installation vom Tar-Archiv:

Wing kann auch vom tar-Archiv installiert werden. Dies kann für Systeme verwendet werden, die kein RPM nutzen oder wenn Sie Wing in einem anderen Verzeichnis als `/usr/lib/wingide-101` installieren möchten. Das Entpacken dieses Archivs mit `tar -zxvf wingide-101-2.1.3-1-i386-linux.tar.gz` wird ein `wingide-101-2.1.3-1-i386-linux` Verzeichnis erstellen, welches das `wing-install.py` Skript und eine `binary-package.tar` Datei enthält.

Die Ausführung des `wing-install.py` Skripts wird nach einem Speicherort für die Installation der Support-Dateien für Wing (`WINGHOME`) und nach einem Speicherort, an dem ein symbolischer Link zu `wing-1012.1` erstellt wird, verlangen. Diese Speicherorte sind auf `/usr/local/lib/wingide-101` beziehungsweise `/usr/local/bin` voreingestellt. Das Installationsprogramm muss Lese-/Schreibzugriff auf beide Verzeichnisse haben und alle Nutzer, die Wing ausführen, müssen auf beide Verzeichnisse Lesezugriff haben.

Der Installationsort von Wing wird als `WINGHOME` bezeichnet. Wenn Sie von tar installiert haben, wird das der Speicherort sein, den Sie auswählen, wenn Sie den Installer ausführen.

Installation auf Debian Linux:

Sie können das Linux-RPM in ein Debian-freundliches Paket umwandeln, indem Sie das `alien` Modul verwenden. So funktioniert's:

1) Das RPM-Paket herunterladen.

2) Das alien-Paket installieren, welches Teil der Debian-Paket-Kollektion ist. Verwenden Sie apt-get, kpackage, aptitude oder einen anderen Paketmanager, um es zu finden und zu installieren.

3) Wechseln Sie zu dem Verzeichnis, in dem das Wing IDE RPM platziert ist, und führen den folgenden Befehl in der Befehlszeile aus:

```
alien -d wingide-101-2.1.3-1.i386.rpm
```

4) Eine `wingide-101-2.1.3-1.deb` Datei ist jetzt im gleichen Verzeichnis zu finden. Um dieses Paket zu installieren führen Sie den folgenden Befehl in der Befehlszeile aus:

```
dpkg -i wingide-101-2.1.3-1
```

Dies ist eine allgemeine Technik, die auch für andere Linux-RPMs funktioniert.

Verwendung des systemweiten GTK:

Wing IDE läuft standardmäßig mit seiner eigenen Kopie von GTK2 und nimmt das systemkonfigurierte Thema nicht auf. Dies wird gemacht, um Probleme und Fehler, die manchmal durch geringe Binärinkompatibilitäten in GTK-Versionen verursacht werden, zu vermeiden.

In Linux-Versionen, die GTK-Version 2.2 oder höher enthalten, können Sie Wing IDE auffordern, das systemdefinierte GTK2 zu verwenden, indem Sie die Einstellung **'System-GTK'_** verwenden oder mit dem `--system-gtk` Argument der Command Line ausführen.

Die Verwendung des systemweiten GTK2 auf diese Weise funktioniert im Allgemeinen ziemlich gut, kann aber aufgrund von Binärinkompatibilitäten in GTK- und verwandten Bibliotheken zu Abstürzen oder Fehlern der Anzeige führen. Wenn Sie die Einstellung setzen und Wing nicht startet, müssen Sie in der Command Line die Option `--private-gtk` bestimmen, um die Einstellung zu überschreiben.

Nicht-ASCII-Dateipfade auf älteren Linux-Systemen:

Einige ältere Linux-Versionen erfordern das Setzen der Umgebungsvarible G_BROKEN_FILENAMES, bevor Wing IDE's Dialog Datei öffnen/speichern mit Dateipfaden, die nicht-ASCII-Zeichen enthalten, richtig funktioniert. Die Umgebungsvariable ist auf einigen Systemen, auf denen sie benötigt wird, bereits eingestellt; dies ist jedoch nicht immer der Fall.

1.9. Wing IDE entfernen

Windows

In Windows verwenden Sie die Schaltfläche `Programme Hinzufügen/Entfernen`, wählen Wing IDE 101 aus und entfernen es.

Linux/Unix

Zum Entfernen einer RPM-Installation in Linux müssen Sie `rpm -e wingide-101` eintippen.

Zum Entfernen einer tar-Archiv-Installation in Linux/Unix, müssen Sie das `wing-uninstall` Skript in `WINGHOME` aufrufen. Dieser Vorgang wird automatisch alle Dateien entfernen, die seit der Installation nicht geändert wurden. Sie werden danach gefragt, ob alle Dateien, die geändert wurden, entfernt werden sollen.

Mac OS X

Um Wing von Mac OS X zu entfernen, müssen Sie einfach seinen Anwendungsordner in den Papierkorb verschieben.

Eine Patch-Installation entfernen

Wenn Sie vorher Patch-Dateien auf Ihre Wing IDE Installation angewendet haben, müssen Sie verbleibende Dateien und Verzeichnisse nach der Deinstallation manuell entfernen. In diesem Fall sollten Sie Ihr **'Verzeichnis der Benutzereinstellungen'__** aufbewahren, welches Lizenz- und Einstellungsinformationen enthält.

1.10. Fehlerbehebung

Dieser Abschnitt beschreibt, was Sie tun können, wenn bei der Installation oder Anwendung von Wing IDE Probleme auftreten.

> Wir sind für Feedback und Fehlerberichte sehr dankbar. Beide können direkt von Wing IDE eingereicht werden, indem Sie die Einträge `Feedback einreichen` und `Fehlerbericht einreichen` aus dem Menü Hilfe verwenden oder uns eine E-Mail an **'support at wingware.com'__** senden.

1.10.1. Fehlerbehebung für Startfehler

Wenn Sie Probleme haben, Wing zum Laufen zu bringen, dann lesen Sie diesen Abschnitt, um Informationen zur Diagnose des Problems zu erhalten.

1) In OS X erfordert Wing, dass Sie einen X11 Server installieren und starten, bevor Sie Wing IDE starten. Siehe das **'OS X How-To'__** für Einzelheiten.

2) In OS X muss außerdem die .tar.gz Datei, in der Wing enthalten ist, mit dem StuffIt Expander oder mit dem gnutar Befehl (nicht mit dem tar Befehl) extrahiert werden. Dies muss aufgrund der Längenbeschränkung von Dateinamen in tar erfolgen.

3) In Windows wird das temporäre Verzeichnis des Nutzers manchmal voll, wodurch das Starten von Wing verhindert wird. Prüfen Sie, ob das Verzeichnis mehr als 65.534 Dateien enhält. Einige Versionen von Acrobat Reader lassen sehr viele Sperrdateien in diesem Verzeichnis. Diese Dateien heißen `Acr-xxxx.tmp`.

4) In Linux kann das Setzen der Einstellung **'System-GTK verwenden'_** dazu führen, dass Wing in einigen Linux-System nicht startet. In diesem Fall müssen Sie die Datei `use-system-gtk` aus Ihrem **'Verzeichnis der Benutzereinstellungen'_** entfernen, Wing starten und die Auswahl des Kontrollkästchens für die Einstellung `System-GTK verwenden` aufheben. Das Qt-Thema verursacht oft Abstürze und bei älteren Linux-Systemen können mit dieser Option Probleme auftreten.

5) Um Probleme mit einer Projektdatei oder Einstellungen auszuschließen, benennen Sie Ihr **'Verzeichnis der Benutzereinstellungen'_** um und starten Wing neu. Wenn dies funktioniert, können Sie Dateien aus dem umbenannten Verzeichnis - eine nach der anderen - herüberkopieren, um das Problem zu isolieren. Sie können auch eine E-Mail an support at wingware dot com schicken, wenn Sie Hilfe benötigen.

6) Unter einem Windows Terminal-Server kann es sein, dass Wing nicht in der Lage ist, die Umgebungsvariablen, die es intern verwendet, zu setzen und wird daher nicht starten. In diesem Fall können Sie Wing mit den folgenden Befehlen zum Laufen bringen:

```
set PYTHONOPTIMIZE=1
set PYTHONHOME=D:\Program Files\WingIDE\bin\PyCore
wing.exe
```

   Ändern Sie `PYTHONHOME` entsprechend dem Speicherort, an dem Sie Wing IDE installiert haben.

7) In anderen Fällen lesen Sie bitte den Abschnitt **'Diagnoseausgabe erhalten'_**.

1.10.2. Probleme in Microsoft Windows

Wing hat einige Probleme/Beschränkungen in Microsoft Windows Systemen.

1) Einige Demo-Shell Erweiterungs-COM-Objekte von win32all können Wing zum Abstürzen bringen, wenn sie registriert sind. Das Abstürzen passiert, wenn die Dialogboxen Datei öffnen, Speichern und Dateien zum Projekt hinzufügen verwendet werden. Diese Erweiterungen können mit ShellExView

(http://www.snapfiles.com/get/shellexview.html) deaktiviert werden; Sie können auch ein ähnliches Programm nutzen, um die Erweiterungen zu finden und zu deaktivieren. Die Erweiterungen können auch deinstalliert werden, indem die .py Datei mit einem `--unregister` Argument ausgeführt wird.

2) Der nVidia Desktop-Manager kann in einigen Windows-Versionen zu Abstürzen führen (die Grafikkarte scheint langsam zu werden, während die Ausnutzung des System-CPU ungefähr 0 % bleibt). Dieses Problem tritt am häufigsten auf, wenn Wing mit mehreren Fenstern verwendet wird, aber kann auch in allen anderen Fällen passieren. Die Deaktivierung des Managers verhindert das Abstürzen.

Es können auch andere Anzeigefehler auftreten (beispielsweise kann es passieren, dass der Fensterinhalt nicht angezeigt wird, wenn das Fenster von der Windows Menüleiste wiederhergestellt wird). Dies ist insbesondere für einige nVidida-Karten der Fall, selbst wenn der Desktop-Manager deaktiviert ist. Wir untersuchen das Problem und arbeiten an der Fehlerbehebung für zukünftige Versionen.

3) Windows Ziehen-und-Ablegen (Drag-n-Drop) funktioniert nicht für die Übertragung von Daten (Text oder Dateien) zwischen Wing und Windows Desktop oder anderen Anwendungen.

1.10.3. Wing IDE beschleunigen

Wing sollte selbst auf relativ langsamer Hardware eine ansprechbare, gut aussehende Benutzeroberfläche präsentieren. In einigen Fällen kann Wing träge erscheinen:

1) Das erste Mal, wenn Sie eine Projektdatei einrichten, analysiert Wing alle Source-Dateien für den Source-Code-Browser und die Auto-Vervollständigungs-Einrichtungen. Während dieser Zeit werden die klassenorientierten Ansichten des Browsers nur die Source-Konstrukte von Dateien, von denen bereits Analyseinformationen erhalten wurden, anzeigen. Die Benutzeroberfläche kann auch träge erscheinen und Wing wird eine wesentliche Menge der CPU-Zeit verbrauchen.

   Um diesen Effekt in nachfolgenden Sitzungen zu begrenzen, speichert Wing seine Source-Analyse-Informationen auf der Festplatte in einem Cache innerhalb Ihres **'Verzeichnisses der Benutzereinstellungen'_**.

   In großen Projekten kann jedoch selbst das Lesen dieses Cache und das Überprüfen von Dateien auf Aktualisierungen eine Weile dauern, wenn Wing das erste Mal gestartet wird. Der Prozess geschieht im Hintergrund nach dem Start und dauert 7-15 Sekunden pro 100.000 Code-Zeilen auf einem Celeron 400 Prozessor.

In allen Fällen wird Wing diesen Prozess schließlich beenden und sollte zu dieser Zeit während normalem Bearbeiten und Debuggen fast kein CPU verbrauchen.

2) In wxPython und anderem Code, der `from xxx import *` Stil-Importe verwendet, kann der Auto-Vervöllständiger anfangs langsam erscheinen, da er viele hundert Symbole verarbeiten muss. Dies sollte jedoch nur das erste Mal, wenn er aufgerufen wird, passieren.

3) Einige Nutzer haben berichtet, dass der Hummingbird Socks Client für Windows ein wesentliches Verlangsamen des Debuggers verursacht, was scheinbar aus falschen Routine-TCP/IP-Paketen resultiert.

1.10.4. Fehlerbehebung öffnungs-Fehler der Dateinamen mit Leerzeichen

In Windows: Wenn Sie Windows Dateiarten oder `Öffnen Mit` verwenden, um zu veranlassen, dass Python-Dateien mit Wing geöffnet werden, setzen einige Versionen von Windows die falsche Befehlszeile für das Öffnen der Datei. Sie können dieses Problem beheben, indem Sie *regedt32.exe*, *regedit.exe* oder ein ähnliches Werkzeug verwenden, um die folgende Registrierungsposition zu bearbeiten:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Applications\wing.exe\shell\open\command
```

Das Problem ist, dass der dort gespeicherten Assoziation Anführungszeichen um das *%1* Argument fehlen. Es sollte stattdessen folgendermaßen lauten:

```
"C:\Programme\Wing IDE\bin\wing.exe" "%1" %*
```

In Linux: KDE's Konqueror hat das gleiche Problem, dass Dateinamen, die von der Befehlszeile an Anwendungen, die an eine Dateiart gebunden sind, weitergegeben werden, nicht von Anführungszeichen umgeben sind, d.h. die Befehlszeile wird nicht korrekt analysiert. Zur Zeit haben wir keine Lösung für dieses Problem.

Anpassung

Es gibt viele Mögichkeiten, Wing IDE auf Ihre Bedürfnisse und Wünsche anzupassen. Dieses Kapitel beschreibt die Optionen, die für die persönliche Anpassung Ihrer Wing IDE Installation zur Verfügung stehen.

Die folgenden Anpassungsmöglichkeiten stehen zur Verfügung:

- Der Inhalt, das Layout und das Aussehen der IDE-Fenster können konfiguriert werden.
- Viele andere Optionen sind in den Einstellungen verfügbar.
- Der Editor kann mit verschiedenen Individualitäten ausgeführt werden (Emacs oder Standard).
- Tastaturkürzel können für jeden beliebigen Wing-Befehl hinzugefügt, entfernt oder geändert werden.
- Datei-Sets können definiert werden, um einige der IDE-Funktionen zu steuern.

2.1. Optionen der Benutzeroberfläche

Wing stellt eine Vielzahl von Optionen bereit, mit denen Sie die Benutzeroberfläche auf Ihre Bedürfnisse anpassen können. Einstellungen können gewählt werden, um die Anzahl und die Art der Fenster, die das IDE verwendet, auszuwählen, das Layout von Werkzeugen innerhalb der Fenster zu bestimmen, die Schriftart und -größe des angezeigten Textes, die Art und den Inhalt der Werkzeugleiste sowie das insgesamte Aussehen oder „Thema“ festzulegen.

2.1.1. Layout der Benutzeroberfläche

Wenn Sie mit der voreingestellten Fensteraufteilung arbeiten, dann besteht der Hauptbereich der Benutzeroberfläche aus zwei Werkzeugboxen (standardmäßig im unteren Teil und auf der rechten Seite, aber dies kann in den **Einstellungen**_ geändert werden) und einem Bereich für Source-Editoren und die integrierte Hilfe.

Das Klicken auf einen bereits aktiven Notizbuchreiter führt dazu, dass Wing das gesamte Feld minimiert, so dass nur noch die Notizbuchreiter sichtbar sind. Ein erneuter Klick bringt die Werkzeugbox wieder zu ihrer ursprünglichen Größe zurück. Die Tasten F1 und F2 wechseln zwischen diesen Modi.

In anderen Fenstermodi werden die Werkzeugboxen und der Editorbereich in separaten Fenstern präsentiert, aber sie haben viele der unten beschriebenen Konfigurationsoptionen gemeinsam.

Konfiguration der Werkzeugleiste

Die Konfigurationsoptionen für Wing's Werkzeugleiste ermöglichen Ihnen, die Größe und die Art der Werkzeugleistensymbole zu ändern. Außerdem können Sie bestimmen, ob zusätzlich oder anstelle der Symbole Text angezeigt werden soll. Sie steuern diese Optionen mit den Einstellungen **'Größe der Werkzeugleistensymbole'**_ und **'Art der Werkzeugleistensymbole'**_.

Die Werkzeugleiste kann auch vollständig versteckt werden. Verwenden Sie dafür die Einstellung **'Werkzeugleiste anzeigen'**_.

Konfiguration des Editorbereiches

Das Popup-Menü Optionen in der oberen rechten Ecke des Editorbereiches ermöglicht Ihnen, den Editor in mehrere unabhängige Felder zu teilen oder mehrere Felder zusammenzufügen. Diese können waagerecht, senkrecht oder in einer beliebigen Kombination von diesen angeordnet werden. Bei mehreren vorliegenden Feldern sind alle innerhalb des Fenters geöffneten Dateien in jedem einzelenen Feld verfügbar. Dies ermöglicht Ihnen, an jeder beliebigen Auswahl von Dateien und/oder in verschiedenen Teilen derselben Datei zu arbeiten.

Das Popup-Menü Optionen kann auch verwendet werden, um zwischen Editoren mit Reitern und Editoren, die ein Popup-Menü für die Auswahl der Dateien anzeigen, zu wechseln (letzteres kann bei einer großen Anzahl von Dateien leichter zu handhaben sein). Außerdem können Sie Editoren in separate Fenster oder zwischen bestehenden Fenstern, wenn mehrere Fenster geöffnet sind, verschieben.

Konfiguratin der Werkzeugboxen

Auch jede Werkzeugbox kann entlang der Axen des Notizbuches in jede beliebige An-

zahl von Unterfeldern geteilt oder zusammengefügt werden. Klicken Sie dafür auf das Drop-Down-Symbol Optionen im Reiterbereich der Notizbücher (ein rechter Mausklick funktioniert auch). Die Anzahl der Werkzeugboxteilungen, die Wing standardmäßig anzeigt, hängt von der Größe Ihres Bildschrims ab.

Das Popdown-Menü Optionen kann auch zur Duplizierung von Werkzeugen oder zum Verschieben von Werkzeugen zwischen den Teilbereichen oder in separate Fenster verwendet werden.

Alle verfügbaren Werkzeuge werden im Menü Werkzeuge aufgezählt. Dieses Menü zeigt das zuletzt verwendete Werkzeug dieser Art an oder es fügt ein Werkzeug zu Ihrem Fenster an seinem voreingestellten Ort hinzu, wenn es noch nicht vorhanden ist.

Zusätzliche Fenster erstellen

Zusätzlich zum Verschieben von bestehenden Editoren oder Werkzeugen in neue Fenster ist es auch möglich, neue Werkzeugfenster (anfänglich mit einem Werkzeug) und neue Dokumentfenster (mit Editor und Werkzeugleiste, wenn es auf die gewählte Fensteraufteilung zutrifft) zu erstellen. Diese Optionen finden Sie im Menü Fenster.

Wing IDE wird den Zustand von all Ihren Fenstern als Teil der Projektdatei speichern, so dass das gleiche Fensterlayout und die gleichen Fensterinhalte in nachfolgenden Sitzungen wiederhergestellt werden.

2.1.2. Änderung der Textanzeige

Wing versucht, für jedes System, auf dem es ausgeführt wird, eine entsprechende Schriftart für die Anzeige zu finden. Viele Nutzer möchten jedoch sicherlich die Schriftart und -größe, die im Editor und anderen Bereichen der Benutzeroberfläche verwendet werden, auf die persönlichen Bedürfnisse anpassen. Sie können dies mit den Einstellungen **'Schriftart/-größe des Source-Codes'_** und **'Schriftart/-größe der Anzeige'_** vornehmen.

Die Konfiguration von Farbe und Schriftart der Syntax-Markierung ist derzeit nicht möglich, ohne dabei den Source-Code des IDE's zu verändern. Wing bietet allerdings die Möglichkeit, die Hintergrundfarbe des Editors einzustellen (mit der Einstellung **'Hintergrund des Source-Codes'_**) und wird entsprechend dem ausgewählten Hintergrund angemessen sichtbare Farben für die Syntax-Markierung bestimmen.

Die Farbe, die für die Textmarkierung verwendet wird, kann auch festgelegt werden. Verwenden Sie dafür die Einstellung **'Farbe der Textmarkierung'_**.

Änderungen der Farbeinstellung hängen oft vom ausgewählten, gesamtheitlichen Anzeigethema ab. Dies wird im nächsten Abschnitt erläutert.

2.1.3. Einstellung des insgesamten Anzeigethemas

Wing IDE basiert auf GTK2, einem auf mehreren Betriebssystemen funktionierendes Benutzeroberflächen-Toolkit, das anpassbare `Themen` bereitstellt, die das insgesamte Look & Feel der Benutzeroberfläche bestimmen. Wing's Standardthema variiert je nach Plattform (in Windows wird ein Windows Emulationsthema verwendet und in OS X wird ein OS X ähnliches Thema genutzt). Das Thema kann mit der Einstellung **'Anzeigethema'__** geändert werden.

In den meisten Fällen wird das neue Thema sofort auf Wing's Benutzeroberfläche angewendet. Wenn Sie zur Standardeinstellung zurückwechseln möchten, kann in einigen Fällen ein Neustart erforderlich sein, was mit einer Nachrichtendialogbox angezeigt wird.

Einige Systeme mit langsameren Grafikkarten können mit den farbreicheren 3D-Themen nicht so gut funktionieren. In diesem Fall ist die Verwendung von `Gtk-Standard` die beste Option, da es keine zusätzliche Grafikverarbeitung umfasst.

System-GTK auf Linux

Auf Linux-Systemen mit GTK 2.2 oder höher ist es möglich, dass Wing mit der systemweiten GTK-Installation und systemdefinierten Themen ausgeführt wird. Dies wird mit der Einstellung **'System-GTK verwenden'__** oder mit den **'Argumenten der Command Line'__** `--system-gtk` sowie `--private-gtk` gesteuert. Wing funktioniert mit den meisten 2.4.x GTK2-Releases ziemlich gut, aber es können trotzdem noch Probleme auftreten. Sollten Sie Probleme mit der Stabilität von Wing haben oder Funktionsstörungen der Anzeige beobachten, empfehlen wir Ihnen, die private GTK-Option zu verwenden.

2.2. Einstellungen

Wing besitzt viele Einstellungen, die die Funktionen des Editors, Debuggers, Source-Browsers, Projektmanagers und anderer Werkzeuge steuern.

Verwenden Sie den Eintrag `Einstellungen` im Menü `Bearbeiten`, um diese Einstellungen zu ändern. Dies ordnet alle verfügbaren Einstellungen nach Kategorie und stellt Zugriff auf die Dokumentation in Werkzeug-Tipps bereit, die angezeigt wird, wenn Sie mit der Maustaste über den beschrifteten Bereich links neben jeder Einstellung fahren. Alle Nicht-Standardwerte, die Sie mit dem **'Einstellungsdialog'__** auswählen, werden in der Einstellungsdatei in Ihrem **'Verzeichnis der Benutzereinstellungen'__** gespeichert.

2.3. Editor-Individualitäten

Die voreingestellte Editor-Individualität für Wing implementiert die allgemein üblichen Tastaturkombinationen, die in einem einfachen, grafischen Texteditor zu finden sind. Diese nutzt für die Interaktion mit dem Editor hauptsächlich die grafische Benutzeroberfläche und begrenzt bei der Interaktion die Verwendung von komplexen, tastaturgesteuerten Befehlen.

Emacs-Individualität
Die erste Sache, die ein Emacs-Nutzer anstrebt, ist es, die Editor-Individualität so einzustellen, dass sie Emacs emuliert. Dies wird mit der Einstellung **'Tastatur / Individualität'__** gemacht.

Mit der Emacs-Individualität können Tastenkombinationen verwendet werden, um den größten Teil der Editor-Funktionalität zu steuern. Es wird eine Dialogzeile für die Textinteraktionen ('Mini-Buffer') am unteren Ende des Editor-Fensters verwendet, in der normalerweise die aktuelle Zeilennummer und andere informative Nachrichten angezeigt werden.

Es ist auch möglich, innerhalb jeder dieser Individualitäten individuelle Tastaturkürzel hinzuzufügen, zu ändern oder zu entfernen. Siehe **'Tastaturkombinationen'__** für Einzelheiten.

Source-Code-Editor

Wing IDE's Source-Code-Editor ist so gestaltet, dass es einfach für Sie ist, mit dem IDE zu arbeiten, selbst wenn Sie an andere Editoren gewöhnt sind.

Editor-Übersicht
Schlüsselelemente, die Sie über den Editor wissen sollten:

- Der Editor hat Individualitäten, einschließlich einer, die Standard-Editoren in Windows ähnlich ist und einer anderen ähnlich zu Emacs.
- Tastaturkombinationen sind konfigurierbar.
- Der Editor unterstützt Syntax-Farbmarkierungen für eine breite Auswahl von Dateitypen.
- Der Editor unterstützt strukturelles Falten für einige Dateitypen.
- Auto-Vervollständigung wird für Python-Source unterstützt.

3.1. Syntax-Farbmarkierung

Der Editor wird versuchen, Dokumente entsprechend ihres MIME-Typen, welcher vom Dateizusatz bestimmt wird, oder entsprechend ihres Inhalts zu markieren. Zum Beispiel wird jede Datei, die mit".py" endet, als ein Python Source-Code-Dokument markiert. Jede Datei, deren MIME-Typ nicht bestimmt werden kann, wird den gesamten Text standardmäßig in schwarzer Normalschrift anzeigen.

Alle verfügbaren Dokumenttypen für Farbmarkierungen sind im Dialog Dateieigenschaften im Editor-Reiter aufgelistet. Wenn Sie mit einer Datei arbeiten, die nicht automatisch erkannt wird, können Sie das Menü Syntax-Markierung verwenden, um die Art, wie diese Datei angezeigt wird, zu ändern. Die Auswahl aus diesem Menü wird in Ih-

rer Projektdatei gespeichert, so dass hier vorgenommene Änderungen im Kontext dieses Projektes dauerhaft sind.

Wenn Sie viele Dateien mit einer unerkannten Erweiterung haben, verwenden Sie die Einstellung **'Extra-Mime-Typen'_**, um Ihre Erweiterung hinzuzufügen.

3.2. Rechtsklick-Menü des Editors

Popup-Menü des Editors
Ein rechter Mausklick auf die Oberfläche des Editors schlägt ein Popup-Menü mit allgemein verwendeten Befehlen, wie Kopieren, Einfügen, Rückgängig und Wiederherstellen, auf. Wenn die Datei eine Python-Datei ist, enthält dieses Menü auch einen Befehl, um zum Punkt der Definition für den Wert, auf den geklickt wurde, zu zoomen.

3.3. Source-Code-Navigation

Das Set von Menüs am Anfang des Editors kann verwendet werden, um durch Ihren Source-Code zu navigieren. Wenn die Reiter des Editor Notizbuches unsichtbar sind, enthaltet die am weitesten links gelegenen Menü eine Liste des geöffneten Dateis. Die zusätzlichen Popup-Menüs zeigen den Bereich der aktuellen Cursor-Auswahl in der Datei an und können verwendet werden, um innerhalb des Top-Level-Bereichs oder innerhalb von Unterbereichen, wenn diese existieren, zu navigieren.

Sie können auch den Menüpunkt `Gehe zur Definition` aus dem Popup-Menü, das mit mit einem rechten Mausklick aufgeschlagen wird, verwenden, um auf ein Konstrukt in Ihrem Source-Code zu klicken und zu dessen Punkt der Definition zu zoomen. Alternativ können Sie den Cursor oder die Auswahl auf einem Symbol platzieren und den Punkt `Gehe zur gewählten Symboldefiniton` aus dem Menü Source oder die entsprechende Tastaturkombination verwenden.

3.4. Dateistatus und nur lesbare Dateien

Die Editor-Reiter oder das Auswahlmenü des Editors (wenn die Reiter versteckt sind) zeigen den Status einer Datei an. Es wird ein `*` angehängt, wenn die Datei bearbeitet wurde, oder `(r/o)` (read-only) hinzugefügt, wenn die Datei nur lesbar ist und nicht geändert werden kann. Diese Information wird für die aktuelle Datei im Statusbereich in der unteren linken Ecke jedes Editor-Fensters gespiegelt.

Dateien, die auf dem Laufwerk nur lesbar sind, werden anfangs in einem nur lesbaren Editor geöffnet. Verwenden Sie das Kontextmenü der Datei (rechter Mausklick), um zwischen dem nur lesbaren Zustand und dem beschreibbaren Zustand zu wechseln. Dies ändert nur die Editierbarkeit des Editors und versucht nicht, den Status der Datei (nur lesbar oder beschreibbar) zu ändern.

3.5. Vorübergehende vs. nicht vorübergehende Editoren

Wing kann Dateien in zwei Modi öffnen:

Modus „Vorübergehend“ -- Dateien, die beim Suchen, Debuggen, Navigieren zum Punkt der Definition sowie bei der Verwendung der Werkzeuge Projekt und Source-Browser (mit dem Kontrollkästchen `Auswahl folgen` aktiviert), geöffnet werden, werden immer im Modus „Vorübergehend“ geöffnet. Diese Dateien werden automatisch geschlossen, wenn sie versteckt werden. Die maximale Anzahl nicht-sichtbarer, vorübergehender Dateien, die jederzeit geöffnet bleiben, kann mit der Einstellung **‘Editor / Erweitert / Schwelle für vorübergehende Dateien‘_** festgelegt werden.

Modus „Nicht vorübergehend“ -- Dateien, die über das Menü Datei, mit der Dateiauswahl über die Tastatur oder mit einem Doppelklick auf Einträge im Projekt-Werkzeug normal geöffnet werden, werdem im Modus „Nicht vorübergehend“ geöffnet. Diese Dateien bleiben solange offen, bis sie ausdrücklich geschlossen werden. Vorübergehende Dateien, die bearbeitet wurden, werden automatisch in nicht vorübergehende Dateien umgewandelt.

Der Modus einer Datei kann zwischen „Vorübergehend“ und „Nicht vorübergehend“ gewechselt werden, indem das Stick-Pin-Symbol in der oberen rechten Ecke des Editor-Bereiches angeklickt wird. Klicken Sie mit der rechten Maustaste auf das Stick-Pin-Symbol, um zwischen den zuletzt besuchten Dateien zu navigieren (blaue Einträge sind vorübergehende Dateien, schwarze Einträge sind nicht-vorübergehend Dateien).

3.6. Klammernsuche

Wing wird zusammenpassende Klammern in grün markieren, wenn der Cursor neben einer Klammer ist. Nicht zusammenpassende Klammern werden in rot markiert.

Sie können Wing dazu veranlassen, die gesamten Inhalte des innersten Klammernpaares von der aktuellen Cursor-Position zu markieren, indem Sie den Eintrag Klammernsuche aus dem Menü Source auswählen.

Für runde Klammern, eckige Klammern und geschweifte Klammern wird in allen Dateien die dazugehörige Klammer gesucht. Bei spitzen Klammern (`<` und `>`) wird auch in HTML- und XML-Dateien die entsprechende Klammer gesucht.

3.6.1. Automatisch Einrücken

Bei der Lieferung des IDE's ist die Funktion Automatisch einrücken angeschalten. Dies verursacht, dass Leerräume am Anfang von jeder neu erstellten Zeile hinzugefügt werden, wenn die Return-Taste oder die Eingabetaste betätigt wird. Es wird genug Leeraum eingefügt, um die Einrückung an das Einrückungslevel der vorherigen Zeile anzupassen und möglicherweise wird ein Einrückungslevel hinzugefügt oder entfernt, wenn dies durch den Kontext des Source-Codes deutlich wird (zum Beispiel `if`, `while` oder `return`).

Beachten Sie, dass wenn die Einstellung **'Automatisch einrücken'_** auf `Falsch` eingestellt ist, erst bei Betätigung der Tab-Taste automatisch eingerückt wird.

3.6.2. Die Tab-Taste

Standardmäßig verhält sich die Tab-Taste genauso wie das automatische Einrücken: Der Leerraum am Anfang der aktuellen Zeile wird angepasst, um ein vernünftiges Einrückungslevel für diese Zeile zu erreichen.

Bestehender Leerraum am Anfang wird durch einen Leerraum ersetzt, der entweder nur Leerzeichen oder Tabs und Leerzeichen enthält, wie durch die oben beschriebene Methode bestimmt. Dieses Verhalten kann auch das Einrückungslevel einer Zeile verringern, wenn es entsprechend seinem Kontext als zu weit eingerückt erachtet wird.

Wenn beim Drücken der Tab-Taste mehrere Zeilen markiert sind, werden alle diese Zeilen als eine Einheit ein- oder ausgerückt, entsprechend der Änderung, die für die erste Zeile der gewählten Einheit notwendig ist. Dies ist sehr hilfreich, wenn Böcke von Code verschoben werden.

Um ein echtes Tabzeichen einzufügen, ungeachtet des Einrückungsmouds oder der Position des Cursors in einer Zeile, tippen Sie Strg-Tab oder Strg-T.

3.6.3. Blockeinrückung ändern

Wing stellt im Einrückungsteil des Menüs Source Befehle zum Einrücken und Ausrücken bereit, um das Erhöhen oder Verringern des Einrückungslevels von markierten Textblöcken zu unterstützen. Alle Zeilen, die in die aktuelle Textauswahl einbezogen sind, werden verschoben, selbst wenn nicht die gesamte Zeile markiert ist.

Einrückungen, die durch diese Befehle gesetzt werden, enthalten entweder nur Leerzeichen, nur Tabs oder eine Mischung aus Tabs und Leerzeichen, wie durch die im Kapitel **'Einrückung'_** beschriebene Methode bestimmt.

3.7. Auto-Vervollständigung

Während Sie Python-Source-Code eingeben, wird Wing ein Popup für die Auto-Vervollständigung anzeigen, das verwendet werden kann, um die Tipparbeit zu reduzieren. Um davon Gebrauch zu machen, tippen Sie solange, bis das korrekte Symbol in der Liste markiert ist und drücken dann die Tab-Taste. Wing wird die verbleibenden Zeichen für das Source-Symbol ergänzen und eventuelle Rechtschreibfehler, die Sie in dem Namen gemacht haben, korrigieren.

Wenn Sie einen Namen auswählen möchten, ohne genug Zeichen einzugeben, die diese Auswahl für den Auto-Vervollständiger eindeutig machen, können Sie auch die Pfeiltasten nach oben und unten auf der Tastatur oder die Maus verwenden, um in der Popup-Liste nach oben oder unten zu rollen. Drücken Sie die Tab-Taste oder doppelklicken Sie auf den Listeneintrag, um das Symbol in Ihrem Source-Code zu vervollständigen.

Um das Popup des Auto-Vervollständigers zu verlassen, klicken Sie auf die `Esc`-Taste oder verwenden Sie `Strg-g`. Der Auto-Vervollständiger wird auch verschwinden, wenn Sie das Source-Symbol verlassen (zum Beispiel indem Sie ein `Leerzeichen` oder irgendein anderes Zeichen, das nicht in einem Source-Symbol enthalten sein kann, drücken) oder wenn Sie andere tastaturgebundene Befehle erteilen, die vom Auto-Vervollständiger nicht akzeptiert werden (zum Beispiel `Speichern` durch die Tastenkombination oder rechte/linke Pfeiltaste).

> **Beschränkungen des Auto-Vervollständigers**
> Die Auto-Vervollständigung deckt momentan die meisten, aber nicht alle möglichen Szenarios ab. Lesen Sie den Abschnitt **'Source-Code-Analyse'_** für zusätzliche Informationen über die gegenwärtigen Fähigkeiten.

3.8. Hinweise zu Kopieren/Einfügen

Es gibt viele Wege, um Text im Editor zu kopieren und einzufügen:

- Verwenden Sie die Einträge des Menüs Bearbeiten. Dies speichert den Text von Kopieren/Ausschneiden in der systemweiten Zwischenablage und kann in andere Anwendungen eingefügt oder von anderen Anwendungen kopiert werden.

- Verwenden Sie die im Menü Bearbeiten definierten Tastenkombinationen.
- Klicken Sie mit der rechten Maustaste auf die Oberfläche des Editors und verwenden die Einträge aus dem Popup-Menü, das erscheint.
- Wählen Sie einen Textbereich und ziehen ihn mit der Funktion 'Ziehen und Ablegen' (Drag and Drop) (das Drücken der Umschalttaste vor dem Ablegen verschiebt den Text anstatt ihn zu kopieren).
- In Linux: Markieren Sie Text irgendwo auf dem Bildschirm und klicken dann mit der mittleren Maustaste, um ihn am Punkt des Klicks einzufügen.
- Im Emacs-Modus: Die Tastenkombination Strg-k (`kill-line`) wird jeweils eine Zeile in die private Emacs-Zwischenablage ausschneiden. Dies wird separat von der systemweiten Zwischenablage gehalten und wird mit der Tastenkombination Strg-y (`yank-line`) eingefügt. In Windows und Mac OS X wird Strg-y die Inhalte der systemweiten Zwischenablage nur dann einfügen, wenn die Emacs-Zwischenablage leer ist.
- In Windows und Mac OS X: Klicken Sie mit der mittleren Maustaste, um die aktuelle, private Emacs-Zwischenablage (wenn in Emacs-Modus und die Ablage ist nicht-leer) oder die Inhalte der systemweiten Zwischenablage (in allen anderen Fällen) einzufügen. In Mac OS X wird die mittlere Maustaste nachgebildet, indem Sie während des Klickens die Programmsteuertaste gedrückt halten.

Es ist wichtig zu beachten, welche Aktionen die systemweite Zwischenablage verwenden, welche die Emacs-Zwischenablage nutzen (nur Emacs-Modus) und welche die X Windows-Auswahl verwenden (nur X Windows). Ansonsten sind diese Befehle in ihren Wirkungen austauschbar.

3.9. Suchen/Ersetzen

Wing stellt eine Vielzahl von Werkzeugen für das Suchen und Ersetzen in Ihrem Source-Code bereit. Welche Sie verwenden, hängt von der Komplexität Ihrer Suchen- oder Ersetzen-Aufgabe ab und davon, mit welchem Stil des Suchens Sie am vertrautesten sind.

3.9.1. Schnellsuche mit der Werkzeugleiste

Eine Möglichkeit, einfache Suchen durchzuführen, besteht darin, Text in das Suchfeld der Werkzeugleiste einzugeben. Während Sie Text eingeben, wird zum nächsten Treffer, der nach der aktuellen Cursor-Position gefunden wird, gerollt. Das Drücken von `Enter`,

sucht jeweils nach dem folgenden Treffer und setzt die Suche am Anfang des Dokumentes fort, wenn das Ende der Datei erreicht ist.

Die Textübereinstimmung bei der Schnellsuche mit der Werkzeugleiste ist von der Groß- und Kleinschreibung unabhängig, es sei denn, Sie geben einen Großbuchstaben als Teil Ihrer Suchzeichenkette ein.

Wenn der Fokus nicht auf dem Suchfeld der Werkzeugleiste liegt und es bereits eine Suchzeichenkette enthält, dann wird, wenn darauf geklickt wird, die Suche nach dem nächsten Treffer sofort im aktuellen Source-Editor starten. Wenn Sie stattdessen nach einer anderen Zeichenkette suchen möchten, dann löschen Sie den Text und geben die gewünschte Suchzeichenkette ein. Während Sie löschen, wird sich die Trefferposition im Editor rückwärts bewegen, bis sie die ursprüngliche Startposition Ihrer Suche erreicht, so dass Ihnen nach dem Eingeben Ihrer neuen Suchzeichenkette der erste Treffer nach der ursprünglichen Cursor-Position des Source-Editors angezeigt wird.

3.9.2. Search Tool

The dockable `Search` tool can be used for more advanced search and replace tasks on the current editor. It provides the ability to customize case sensitivity and whole/part word matching, search in selection, and perform wildcard or regex search and replace.

To the right of the `Search` and `Replace` fields, Wing makes available a popup that contains a history of previously used strings, options for inserting special characters, and an option for expanding the size of the entry area.

The following search options can be selected from the tool:

- **Case Sensitive** -- Check this option to show only exact matches of upper and lower case letters in the search string.
- **Whole Words** -- Check this option to require that matches are surrounded by white space (spaces, tabs, or line ends).
- **In Selection** -- Search for matches only within the current selection on the editor.

The following addittional options are available from the Options popup menu:

- **Show Replace** -- Whether or not the Replace fields are visible on the tool.
- **Text Search** -- Select this to do a regular text search without wildcard or regex.
- **Wildcard Search** -- Select this to allow use of special characters for wildcarding in the search string:

- * can be used to match any sequence of characters except for line endings. For example, the search string `my*value` would match anything within a single line of text starting with `my` and ending with `value`. Note that * is „greedy“ in that `myinstancevalue = myothervalue` would match as a whole rather than as two matches. To avoid this, use Regex Search instead with `.*?` instead of *.
- ? can be used to match any single character except for line endings. For example, `my???value` would match any string starting with `my` followed by three characters, and ending with `value`.
- [and] can be used to indicate sets of match characters. For example `[abcd]` matches any one of `a`, `b`, `c`, or `d`. Also, `[a-zA-Z]` matches any letter in the range from `a` to `z` (inclusive), either lower case or uppercase. Note that case specifications in character ranges will be ignored unless the `Case Sensitive` option described above is turned on.

- **Regex Search** -- Select this to use regular expression style searching. This is a more powerful variant than wildcard search that allows for more complex specification of search matches and replacement values. For information on the syntax allowed for the search and replace strings, see Python's **'Regular Expression Syntax'__** documentation.

- **Wrap Search** -- Uncheck this to avoid wrapping around when the search reaches the top or bottom of a file.
- **Incremental** -- Check this to immediately start or restarted searching as you type or alter search options. When unchecked, use the forward/backward search buttons to initiate searching.
- **Find After Replace** -- Select this to automatically find the next search match after each Replace operation.

3.10. Source-Code-Analyse

Wing's Auto-Vervollständiger, Source-Index-Menü, Gehe-zu-Definition Fähigkeiten, einige Funktionen zur Source-Neuformatierung sowie in Wing IDE Professional der Source-Code-Browser und Source-Assistent verlassen sich alle auf eine zentrale Maschine, die Ihren Source-Code im Hintergrund liest und analysiert, während Sie Dateien zu Ihrem Projekt hinzufügen oder Ihren Code im Source-Code-Editor ändern.

So funktioniert die Analyse
Für die Analyse Ihres Source-Codes wird Wing den Python-Interpreter und den PYTHONPATH, den Sie in Ihren **'Projekteigenschaften'__** bestimmt haben, verwenden. Wenn Sie für Ihr Projekt eine Haupt-Debug-Datei festgelegt haben, dann werden die Eigenschaftswerte dieser Datei verwendet; andernfalls werden die projektweiten Werte verwendet. Wann immer sich irgendeiner dieser Werte ändert, wird Wing Ihren Source-Code vollständig neu analysieren.
Sie können den Python-Interpreter und PYTHONPATH, die von der Source-Code-Analyse-Maschine verwendet werden, ansehen, indem Sie den Eintrag `Analysestatistik anzeigen` aus dem Source-Menü auswählen. Die Werte in dem sich aufschlagenden Dialogfenster sind nur lesbar, aber wenn Sie auf die Schaltfläche Einstellungen klicken, können Sie Änderungen vornehmen. Siehe **'Projektweite Eigenschaften'__** um Einzelheiten darüber zu erfahren, wie Sie diese Werte ändern.
Seien Sie sich bewusst, dass Wing bei der Verwendung von mehreren Versionen des Python-Interpreters oder unterschiedlicher PYTHONPATH-Werte für verschiedene Source-Dateien in Ihrem Projekt alle Dateien im Projekt analysieren wird und die Interpreter-Version und den PYTHONPATH verwenden wird, die es in der Haupt-Debug-Datei oder den projektweiten Debug-Eigenschaften findet. Dies kann zu fehlerhaften oder unvollständigen Analysen einiger Source-Dateien führen. Daher ist es das Beste, nur eine Python-Version mit jeder Wing IDE Projektdatei zu nutzen.

Die folgenden Punkte sind bekannte Beschränkungen, die Funktionen, welche auf der Source-Analyse basieren, beeinflussen:

- Die Analyse scheitert manchmal daran, den Typ eines Konstruktes zu identifizieren, weil der Python-Code nicht immer Anhaltspunkte zur Bestimmung des Datentyps bereitstellt. In diesen Fällen können Sie `isinstance` und/oder Interface-Dateien verwenden, um das Analyseprogramm zu informieren, wie weiter unter beschrieben.
- Typen von Elementen in Listen, Tuples und Dictionaries sind nicht identifiziert.
- Doc-Strings und andere Analyseinformationen können veraltet sein, wenn Sie eine Datei extern mit einem anderen Editor bearbeiten und diese in Wing nicht neu laden. Siehe Abschnitt **'Geänderte Dateien automatisch Neuladen'__** für Optionen zum Neuladen.
- Einige neuere Python-Sprachenkonstrukte und mögliche Fälle der Typenanalyse werden nicht ausdrücklich unterstützt.

Verwendung von isinstance() zur Unterstützung der Analyse
Eine Möglichkeit, die Einrichtung der Code-Analyse über den Typ einer Variablen zu informieren, besteht darin, einen `isinstance`-Aufruf zu Ihrem Code hinzuzufügen. Ein Beispiel ist `assert isinstance(obj, CMyClass)`. Das Code-Analyseprogramm wird diese aufnehmen und vollständigere Informationen für diese Werte bereitstellen.

Die Verwendung von *.pi Dateien zur Unterstützung der Analyse
Wing's Source Analyser kann nur Python-Code lesen und umfasst keinen Support zum Verstehen für den Code von C/C++ Erweiterungsmodulen. Um die Code-Analyse über die Inhalte eines Erweiterungsmoduls zu informieren, ist es möglich, eine `*.pi` (Python-Interface) Datei zu erstellen. Zum Beispiel wird die Interface-Datei für ein Modul, das als `mymodule` importiert wird, als `mymodule.pi` bezeichnet. Diese Datei ist einfach ein Python-Skeleton mit der entsprechenden Struktur und Call-Signatur, damit sie mit den Funktionen, Attributen, Klassen und Methoden, die in einem Erweiterungsmodul definiert sind, übereinstimmt. In vielen Fällen können diese Dateien aus den Interface-Dateien automatisch erzeugt werden.
Wing sucht nach `*.pi` Dateien zuerst im gleichen Verzeichnis, in dem es das Erweiterungsmodul findet (oder im Verzeichnis des Source-Codes, wenn das Modul noch nicht kompiliert wurde und sich das Verzeichnis des Source-Codes in Ihrem konfigurierten Python Path befindet). Wenn die Dateien nicht gefunden werden, sucht Wing im Verzeichnispfad, für den die Einstellung **'Schnittstellenpfad'**_ gesetzt ist. Schließlich wird Wing im Verzeichnis `resources/builtin-pi-files` innerhalb Ihrer Wing IDE Installation suchen.
Bei der Suche im Schnittstellen-Pfad oder in Wing's Builtin-Verzeichnis wird zuerst die höchste Ebene des Verzeichnisses nach einer übereinstimmenden `*.pi` Datei durchsucht. Danach sucht Wing in einem Unterverzeichnis `#.#`, das entsprechend der Haupt- und Unterversion von Python, das mit Ihrer Source-Basis verwendet wird, benannt ist. Im Folgenden wird dann jede niedrigere Haupt-/Unterversion rückwärts bis `1.5` durchsucht.
Wenn sich zum Beispiel `c:\share\pi\pi-files` im Schnittstellenpfad befindet und Python 2.3 verwendet wird, sucht Wing zuerst in `c:\share\pi\pi-files`, dann in `c:\share\pi\pi-files\2.3`, danach in `c:\share\pi\pi-files\2.2` und so weiter.
Beispiele für `*.pi` Dateien, die von Wing intern für die Erstellung von Auto-Vervollständigungsinformationen für Builtins verwendet werden, sind in dem Verzeichnis `resources/builtin-pi-files` innerhalb Ihrer Wing IDE Installation zu finden. Dieses veranschaulicht auch den oben beschriebenen Rückwärtsmechanismus der Versionsnummern.

3.10.1. Analyse-Cache

Der Source-Code-Analyser speichert Informationen über Dateien, die er kürzlich geprüft hat, unter `cache` in Ihrem **'Verzeichnis der Benutzereinstellungen'__**.

Die Größe des Cache-Speichers kann mit der Einstellung **'Maximale Cache-Größe'__** kontrolliert werden. Wing bringt allerdings keine so gute Leistung, wenn der für den Cache-Speicher verfügbare Raum kleiner ist als der Raum, der für die Source-Analyse-Informationen eines einzelnen Projekts benötigt wird. Wenn Sie extreme Verlangsamungen bemerken, erhöhen Sie entweder die Größe des Cache-Speichers oder deaktivieren Sie ihn vollständig, indem Sie seine Größe auf 0 setzen.

Wenn der Cache-Speicher von mehr als einem Computer verwendet wird, versichern Sie sich, dass die Uhren der beiden Computer synchronisiert sind. Der Cache-Mechanismus verwendet Zeitstempel und kann verwirrt werden, wenn dies nicht gemacht wird.

Der Analyse-Cache kann in seiner Gesamtheit ohne negative Auswirkungen entfernt werden.

Debugger

Wing's Debugger stellt ein leistungsfähiges Werkzeugset für die schnelle Lokalisierung und Behebung von Fehlern in Python-Code bereit. Er unterstützt Haltepunkte, das Schreiten durch den Code, die Prüfung und Änderung von Stack- oder Moduldaten, Watchpoints, Ausdrucksbewertung und die Interaktion im Command-Shell-Stil mit dem angehaltenen Debug-Prozess.

Der Debugger ist um ein TCP/IP Client/Server-Design erstellt, welches das Starten Ihrer Anwendung nicht nur von Wing selbst, sondern auch extern, wie mit CGI-Skripten oder Code, der in einer eingebetteten Skripting-Einrichtung innerhalb einer größeren Anwendung läuft, unterstützt. Remote-Debuggen (Host zu Host) steht auch zur Verfügung.

Da der Debugger-Kern in optimiertem C geschrieben ist, ist der Debug-Overhead relativ niedrig. Sie sollten jedoch damit rechnen, dass Ihre Programme innerhalb des Debuggers etwa 50% langsamer laufen.

4.1. Haltepunkte setzen

Haltepunkte können im Source-Code gesetzt werden, indem Sie die Source-Datei öffnen und links von einer Source-Code-Zeile auf den Haltepunktrand klicken. Alternativ können das Menü Debuggen oder die Haltepunktsymbole aus der Werkzeugleiste verwendet werden, um Haltepunkte an der aktuellen Zeile des Source-Codes (wo der Einfügecursor oder die Markierung ist) zu setzen oder zu löschen.

4.2. Debuggen starten

Es gibt mehrere Wege, eine Debug-Sitzung innerhalb von Wing zu starten:

- Wählen Sie `Debuggen / Fortsetzen` aus dem Menü Debuggen oder klicken Sie auf das Symbol `Debuggen` in der Werkzeugleiste. Das wird die Haupt-Debug-Datei,

wenn eine bestimmt ist (beschrieben im Abschnitt **‘Eine Haupt-Debug-Datei setzen‘_**), oder andernfalls die im vordersten Editor-Fenster geöffnete Datei ausführen. Die Ausführung hält am ersten Haltepunkt oder der ersten Exception an oder stoppt nach der Prgrommbeendigung.

- Wählen Sie `In Funktion` aus dem Menü Debuggen oder klicken Sie auf das Symbol `In Funktion` in der Werkzeugleiste. Dies wird die Haupt-Debug-Datei, wenn eine bestimmt ist, oder andernfalls die im vordersten Editor-Fenster geöffnete Datei ausführen. Die Ausführung stoppt an der ersten Code-Zeile.
- Wählen Sie `Aktuelle Datei debuggen` aus dem Menü Debuggen oder `Ausgewählte Debuggen` aus dem Popup-Menü, das mit einem rechten Mausklick auf das Projektwerkzeug aufgeschlagen wird, um eine spezifische Datei auszuführen, unabhängig davon, ob für Ihr Projekt eine Haupt-Debug-Datei bestimmt wurde. Dies wird am ersten Haltepunkt oder an der ersten Exception stoppen oder es wird nach Beendigung des Programms angehalten.
- Wählen Sie `Gehe zum Cursor` aus dem Menü Debuggen. Dies wird die Haupt-Debug-Datei, wenn eine bestimmt ist, oder andernfalls die im vordersten Editor-Fenster geöffnete Datei ausführen. Die Ausführung wird solange fortgesetzt bis sie die im aktuellen Source-Text-Fenster markierte Zeile erreicht, bis sie auf einen Haltepunkt oder eine Exception trifft oder bis das Programm beendet ist.
- Verwenden Sie `Letzte Debuggen` aus dem Menü Debuggen, um eine kürzlich gedebuggte Datei auszuwählen. Dies wird am ersten Haltepunkt oder an der ersten Exception stoppen oder nach Beendigung des Programms anhalten.
- Verwenden Sie einen der Tastaturbefehle, die im Menü Debuggen zu finden sind. Im Emacs-Modus ist der Tastaturbefehl `Strg-C Strg-C` auch implementiert.

Zusätzliche Optionen bestehen für das Starten einer Debug-Sitzung von außerhalb von Wing IDE und für das Anhängen an einen bereits laufenden Debug-Prozess. Diese sind in den Abschnitten **‘Extern gestarteten Code debuggen‘_** und **‘Anhängen‘_** beschrieben.

Sobald ein Debug-Prozess gestartet wurde, sollte sich das Statuslicht in der oberen rechten Ecke des Werkzeuges Stack-Daten von rot auf eine andere Farbe ändern, wie in **‘Debugger-Status‘_** beschrieben.

4.3. Debugger-Status

Die Debugger-Werkzeuge Stack-Daten, Beobachten und Debug-Test enthalten eine Statusanzeige, die sich rechts neben dem Popup-Menü für die Stack-Auswahl befindet. Die

Statusanzeige kann verwendet werden, um den Zustand des Debuggers folgendermaßen zu bestimmen:

- **Reines Rot** -- Es existiert kein Debug-Prozess und der Debugger hört nicht auf Verbindungen.
- **Rot mit Schrägstrich** -- Es existiert kein Debug-Prozess, aber der Debugger hört auf Verbindungen von extern gestarteten Prozessen.
- **Gelb** -- Ein Debug-Prozess ist angehängt und wird ausgeführt oder er ist beim Anhängen.
- **Grün** -- Ein Debug-Prozess ist angehängt und an einem Haltepunkt oder einer Exception angehalten oder gestoppt.

Wenn Sie mit der Maus über die Statusanzeige fahren, wird ein Werkzeug-Tipp angezeigt, der den Debugger-Status beschreibt.

Der aktuelle Status des Debuggers ist außerdem im Nachrichtenwerkzeug des IDE's in der Debugger-Statusgruppe aufgelistet.

4.4. Ablaufsteuerung

Wenn der Debugger einmal läuft, sind die folgenden Befehle zur Kontrolle der weiteren Ausführung des Debug-Programms von Wing verfügbar. Diese können über die Werkzeugleiste oder das Menü Debuggen erreicht werden:

- Ein frei-laufendes Debug-Programm kann jederzeit mit der Option `Anhalten` aus dem Menü Debuggen oder mit der Schaltfläche Anhalten aus der Werkzeugleiste angehalten werden. Dies wird am aktuellen Punkt der Ausführung des Debug-Programms anhalten.
- Während einer Debug-Sitzung kann jederzeit der Menüeintrag oder das Werkzeug `Debuggen Stoppen` verwendet werden, um die Beendigung des Debug-Programms zu erzwingen.

  Diese Option ist standardmäßig deaktiviert, wenn der aktuelle Prozess außerhalb von Wing gestartet wurde. Sie kann für alle lokalen Prozesse mit der Einstellung **'Externe Löschen aktivieren'_** aktiviert werden.

Wenn an einer gegebenen Code-Zeile gestoppt wurde, kann die Ausführung mit dem Menü Debuggen oder der Werkzeugleiste wie folgt kontrolliert werden:

- **Über Funktion** schreitet über eine einzelne Zeile von Python-Code.
- **In Funktion** wird versuchen, in die nächste ausgeführte Funktion in der aktuellen Code-Zeile zu gehen. Wenn dort keine Funktion oder Methode ist, in die hineingegangen werden kann, dann verhält sich dieser Befehl wie Aus Funktion.
- **Aus Funktion** wird die Ausführung der aktuellen Funktion oder Methode abschließen und an der ersten Anweisung, die nach der Rückkehr von der aktuellen Funktion oder Methode angetroffen wird, stoppen.
- **Fortsetzen** wird die Ausführung bis zum nächsten Haltepunkt, zur nächsten Exception oder zum Programmende fortsetzen.
- **Gehe zum Cursor** wird zur Stelle des Cursors im vordersten Editor oder zum nächsten Haltepunkt, zur nächsten Exception oder zum Programmende gehen.

4.5. Stack anzeigen

Immer wenn das Debug-Programm an einem Haltepunkt oder während dem manuellen Schreiten anhält, wird der aktuelle Stack im Popup-Menü am Anfang des Werkzeuges Stack-Daten angezeigt. Dies zeigt alle Stack-Frames, die zwischen der Anforderung des Programms und der aktuellen Ausführungsposition angetroffen werden, an. Die äußeren Stack-Frames sind in der Liste weiter oben.

Beachten Sie, dass der angezeigte Stack eine Verkettung von allen gesehenen Python Stack-Frames ist und Unstetigkeiten enthalten kann, wenn Ihr Code C/C++ oder anderen nicht-Python-Code aufruft, der im Gegenzug in Python zurückruft. In diesem Fall werden die C/C++ Stack-Frames fehlen, aber die insgesamte Reihenfolge und der Fluss der Anforderung sollten von denjenigen Stack-Frames, die sichtbar sind, offensichtlich sein.

Wenn der Debugger an einen Haltepunkt oder eine Exception geht oder dort stoppt, wählt er standardmäßig den innersten Stack-Frame aus.

Um andere Stack-Frames weiter oben oder unter im Stack zu besuchen, wählen Sie sie aus dem Popup-Menü der Stack-Daten aus, verwenden die Einträge `Aufwärts Stack` und `Abwärts Stack` aus dem Menü Debuggen oder klicken auf die Symbole Aufwärts/Abwärts in der Werkzeugleiste.

Wenn Sie Stack-Frames wechseln, werden die Variablenansichten entsprechend geändert und die aktuelle Code-Zeile an diesem Stack-Frame wird in einem Editor-Fenster angezeigt.

Beachten Sie, dass der aktuelle Stack-Frame auch verwendet wird, um den Bewertungskontext in den Werkzeugen Debug-Test und Beobachten zu steuern.

4.6. Debug-Daten anzeigen

Der Wing IDE Debugger stellt mehrere Möglichkeiten bereit, mit denen Sie die Daten Ihres Debug-Programms ansehen können:

(1) Durch die Prüfung von Lokalen und Globalen unter Verwendung des Werkzeuges Stack-Daten. Dieser Bereich zeigt Werte für den gegenwärtig gewählten Stack-Frame an.

(2) Durch das Durchsuchen von Werten in allen geladenen Modulen (wie von `sys.modules` bestimmt), unter Verwendung des Werkzeuges Module.

(3) Durch das Beobachten spezifischer Werte von einer der oben genannten Ansichten (klicken Sie mit der rechten Maustaste auf Werte, um sie zum Werkzeug Beobachten hinzuzufügen).

(4) Durch das Eingeben von Ausdrücken in das Beobachten-Werkzeug.

Auf Anfrage abgerufene Werte
Die von Wing angezeigten Variablendaten werden vom Debug-Server schnell abgerufen während Sie navigieren. Aus diesem Grund können Sie eine kurze Verzögerung bemerken, wenn eine Änderung in einer Erweiterung oder einem Stack-Frame zu einer großen Datenübertragung führt.
Aus dem gleichen Grund können große Mengen von Debug-Daten, die auf dem Bildschirm sichtbar bleiben, das Schreiten durch den Code verlangsamen.

4.6.1. Ansicht der Stack-Daten

Das Debugger-Werkzeug Stack-Daten enthält ein Popup-Menü für den Zugriff auf den aktuellen Debug-Stack, einen Baumansichtsbereich für das Durchsuchen von Variablendaten in Lokalen und Globalen und einen Textansichtsbereich für die Prüfung von großen Datenwerten, die in der Baumansicht abgeschnitten sind.

Einfache Werte, wie Strings und Zahlen, und Werte mit einer kurzen String-Ansicht werden in der Wertspalte des Baumansichtbereichs angezeigt.

Strings sind immer in `""` (Anführungszeichen) eingebettet. Jeder Wert außerhalb der Anführungszeichen ist eine Zahl oder eine intern definierte Konstante, wie `None` oder `Ellipsis`.

Ganzzahlen können entweder als dezimal, hexadezimal oder oktal angezeigt werden, was mit der Einstellung **'Anzeigemodus für Ganzzahlen'__** geregelt werden kann.

Komplexe Werte, wie Instanzen, Lists und Dictionaries, werden in eckigen Klammern und mit einer Speicheradresse dargestellt (zum Beispiel `<dict 0x80ce388>`) und können durch Klicken auf die Erweiterungsanzeige in der Spalte Variable erweitert werden. Die Speicheradresse identifiziert das Konstrukt eindeutig. Wenn Sie die gleiche Adresse an zwei Stellen sehen, dann betrachten Sie zwei Objektverweise zu der gleichen Instanz.

Die in den Bereich einer Klasse gehörenden Werte, die innerhalb einer Instanz gesehen werden, sind kursiv angezeigt.

Nach der Erweiterung komplexer Ansichten wird die Position oder der Name von jedem Untereintrag in der Spalte Variable angezeigt und der Wert von jedem Eintrag (möglicherweise auch komplexe Werte) werden in der Spalte Wert angezeigt. Verschachtelte, komplexe Werte können unbestimmt erweitert werden, selbst wenn dies zu einem Traversal der Zyklen der Objektverweise führt.

Wenn Sie einen Eintrag einmal erweitern, wird der Debugger diesen Eintrag weiterhin als erweitert darstellen, selbst nachdem Sie weiter gehen oder die Debug-Sitzung neu starten. Der Erweiterungsstatus wird für die Dauer Ihrer Wing IDE Sitzung gespeichert.

Wenn der Debugger auf einen langen String trifft, wird dies in der Spalte Wert durch das Voranstellen von `...` vor dem abgeschnittenen String angezeigt. In diesen Fällen kann der vollständige Wert des Strings im Textansichtsbereich unten im Stack-Daten-Werkzeug angesehen werden, nachdem in der Baumansicht auf den abgeschnittenen String geklickt wurde.

Unlesbare Daten
Einige Datentypen, wie zum Beispiel solche, die nur innerhalb von C/C++ Code definiert sind oder solche, die bestimmte Internals der Python-Sprache beinhalten, können nicht über das Netzwerk übertragen werden. Diese sind mit Werteinträgen in der Form `<opaque 0x80ce784>` bezeichnet und können nicht weiter erweitert werden. In Wing IDE Professional können Sie allerdings den **'Debug-Test'__** verwenden, um auf sie zuzugreifen (versuchen Sie zum Beispiel `dir(value)` einzutippen).

4.6.1.1. Optionen des Popup-Menüs

Ein rechter Mausklick auf die Oberfläche der Stack-Datenansicht schlägt ein Popup-Menü mit Optionen für die Navigation von Datenstrukturen auf:

- **Mehr Erweitern** -- Wenn ein komplexer Datenwert ausgewählt ist, wird dieser Menüeintrag eine zusätzliche Ebene in dem komplexen Wert erweitern. Da dies eine potentiell große Anzahl von Werten erweitert, werden Sie wahrscheinlich eine Verzögerung bemerken, bis dieser Vorgang abgeschlossen ist.

- **Mehr Zusammenklappen** -- Wenn ein komplexer Datenwert ausgewählt ist, wird dieser Menüeintrag dessen Anzeige um eine zusätzliche Ebene zusammenklappen.
- **Nach ... beobachten** -- Diese Punkte können verwendet werden, um einen Debug-Datenwert über die Zeit zu beobachten, wie in **'Werte verfolgen'__** beschrieben.
- **Neuladen erzwingen** -- Dies zwingt Wing IDE, den angezeigten Wert vom Debug-Prozess neuzuladen. Dies ist in Fällen hilfreich, in denen Wing einen Bewertungsfehler anzeigt oder wenn das Debug-Programm Instanzen enthält, die `__repr__` oder ähnliche besondere Methoden in einer Art implementieren, die verursacht, dass sich der Wert ändert, wenn er der wiederholten Bewertung unterliegt.

4.6.2. Probleme bei der Behandlung von Werten

Der Wing Debugger versucht, Debug-Daten so sanft wie möglich zu behandeln, um das Eintreten von sehr langen Berechnungen oder das Auslösen von Fehlern im Debug-Prozess während dem Packen von Debug-Daten für die Übertragen zu vermeiden. Trotzdem können nicht alle Debug-Daten in der Anzeige angezeigt sein. Dieser Abschnitt beschreibt alle Gründe, warum dies passieren kann:

- **Wing kann bei der Behandlung eines Wertes abschalten** -- Große Datenwerte können den Debug-Server-Prozess während dem Packen aufhängen. Wing versucht dies zu vermeiden, indem es die Größe eines Objektes vor dem Packen sorgfältig testet. In einigen Fällen funktioniert das nicht und Wing wird auf die Daten für die Dauer, die in der Einstellung **'Netzwerkabschaltung'__** festgelegt ist, warten und wird dann den Variablenwert als `<Netzwerkabschaltung während der Bewertung>` anzeigen.
- **Wing kann auf Werte treffen, die für die Verarbeitung zu groß sind** -- Wing wird große Sequenzen, Bereiche oder Strings, die die von den Einstellungen **'Große Listenschwelle'__** und **'Große Stringschwelle'__** festgelegten Größenlimits übersteigen, nicht verpacken und übertragen. In der Debugger-Anzeige werden übergroße Sequenzen und Bereiche als `riesig` kommentiert und `<abgeschnitten>` ist großen, abgeschnittenen Strings vorangestellt.

  Erhöhen Sie die Werte der Schwelleneinstellungen, um dies zu vermeiden; seien Sie aber auf längere Datenübertragungszeiten vorbereitet. Beachten Sie, dass eine zu hohe Einstellung dieser Werte den Debugger zum Abschalten veranlassen wird, wenn der Wert **'Netzwerkabschaltung'__** nicht auch erhöht wird.

- **Wing kann während der Datenbearbeitung auf Fehler treffen** -- Da Wing während dem Packen von Debug-Daten Zuweisungen und Vergleiche vornimmt, und weil es Debug-Daten in String-Form umwandelt, kann es besondere Methoden, wie `__cmp__` und `__str__` in Ihrem Code ausführen. Wenn dieser Code Fehler enthält, kann der Debugger diese Fehler manchmal aufdecken, die Sie ansonsten nicht sehen würden.

  Im schlimmsten Fall, der allerdings nur selten eintritt, wird der Debug-Prozess abstürzen, wenn beschädigter C oder C++ Erweiterungsmodul-Code aufgerufen wird. In diesem Fall wird die Debug-Sitzung beendet.

  Allgemein bekannter, aber trotzdem noch selten, sind Fälle, in denen Wing während der Bearbeitung eines Debug-Datenwertes auf eine unerwartete Python-Exception trifft. Wenn dies passiert, zeigt Wing den Wert als `<Fehler bei Wertbearbeitung>` an.

  Diese Fehler werden im Exceptions-Werkzeug nicht als normale Programmfehler berichtet. Zusätzliche Ausgabe, welche die aufgetretene Exception enthalten kann, kann jedoch durch das Setzen der Einstellung **'Protokolldatei der Internals debuggen'__** erhalten werden.

Wing merkt sich Fehler, auf die es in Debug-Daten trifft, und speichert diese in der Projektdatei. Diese Werte werden während nachfolgendem Debuggen nicht neu abgerufen, selbst wenn Wing beendet und neu gestartet wird.

Um dieses Verhalten für einen einzelnen Wert außer Kraft zu setzen, verwenden Sie den Menüpunkt `Neuladen erzwingen` aus dem mit einem rechten Mausklick aufzuschlagenden Popup-Menü eines Variablenbereiches in Baumansicht.

Verwenden Sie den Eintrag `Gespeicherte Wertefehler löschen` aus dem Menü `Debuggen`, um die Liste aller vorher angetroffenen Fehler zu löschen, so dass alle Werte neu geladen werden. Dies funktioniert nur für die Liste der Fehler, die in der aktuellen Debug-Datei bekannt sind, wenn eine Debug-Sitzung aktiv ist oder für die Haupt-Debug-Datei, wenn vorhanden, wenn kein Debug-Prozess läuft.

4.7. Interaktive Python-Shell

Eine Python-Shell wird für die Ausführung von Befehlen und die Bewertung von Ausdrücken außerhalb Ihres Debug-Programms bereitgestellt.

Da diese Shell einen separaten Python-Prozess, der von Ihrem Debug-Prozess unabhängig ist, ausführt, ist sie immer aktiviert und funktioniert ohne Rücksicht auf den Status eines laufenden Debug-Prozesses.

Die Python-Shell läuft immer mit der gleichen Python-Version, wie die, die auch für Ihren Debug-Prozess verwendet wird. Dies wird genauer im Abschnitt **'Debug-Eigenschaften'__** beschrieben.

Um den Status einer Python-Shell aufzuheben, drücken Sie auf die Schaltfläche `Neue Sitzung`. Dies wird den externen Python-Prozess beenden und ihn neu starten, also den Status der Shell aufheben und neu einstellen.

4.8. Debug-Prozess-I/O

Während Sie unter dem Wing Debugger ausführen, wird der gesamte Verkehr zu und von Python `stdin` und `stdout` und alle Aufrufe zu `input()` und `raw_input()` durch die Debug-Server-Maschinerie umgeleitet. Dieser Code macht zwei Dinge: (1) Alle Warteaufrufe für `sys.stdin` werden im Multiplex-Betrieb ausgeführt, wobei `sys.stdin` und das Debug-Netzwerk-Socket gleichzeitig bedient werden, so dass der Debug-Prozess zu Wing IDE ansprechbar bleibt, während auf Tastatureingaben gewartet wird, und (2) in manchen Fällen wird I/O zu einem anderen Fenster umgeleitet.

Für Debug-Prozesse, die von Wing aus gestartet werden, erscheint der I/O der Tastatur immer im Werkzeug Debug-I/O oder in einer neuen, externen Konsole, die vor dem Starten des Debug-Prozesses erstellt wird. In **'Externe I/O-Konsolen'__** ist beschrieben, wie dies gesteuert werden kann.

Debug-Prozesse, die außerhalb von Wing unter Verwendung von `wingdbstub` gestartet werden, nehmen Ihren Tastatur-I/O immer durch die Umgebung, von der Sie gestartet wurden, vor (dies kann ein Konsolen-Fenster, ein Web-Server oder jede andere I/O-Umgebung sein).

Wenn Befehle in den **'Debug-Test'__** eingegeben werden, wird der I/O während der Zeit, in welcher der Befehl verarbeitet wird, vorübergehend zum Debug-Test umgeleitet.

Lizenzinformationen

Wing IDE ist ein kommerzielles Produkt, das auf einer Reihe von Open Source Technologien basiert. Obwohl der Source-Code des Produktes für Nutzer von Wing IDE Professional zur Verfügung steht (mit Unterzeichnung einer Geheimhaltungsvereinbarung), ist das Produkt selbst nicht Open Source.

Die folgenden Abschnitte beschreiben die Lizenzierung für das Produkt als Ganzes (Endnutzervereinbarung) und stellen die erforderlichen Legal Statements für die enthaltenen Open Source Komponenten bereit.

5.1. Wing IDE Software-Lizenz

This End User License Agreement (EULA) is a CONTRACT between you (either an individual or a single entity) and Wingware, which covers your use of „Wing IDE 101“ and related software components. All such software is referred to herein as the „Software Product.“ A software license and a license key or serial number („Software Product License“), issued to a designated user only by Wingware or its authorized agents, is required for each concurrent user of the Software Product. If you do not agree to the terms of this EULA, then do not install or use the Software Product or the Software Product License. By explicitly accepting this EULA you are acknowledging and agreeing to be bound by the following terms:

1a. EVALUATION LICENSE WARNING

This Software Product can be used in conjunction with a free evaluation Software Product License. If you are using such an evaluation Software Product License, you may use the Software Product only to evaluate its suitability for purchase. Evaluation Software Product Licenses have an expiration date and most of the features of the software will be disabled after that date. WINGWARE BEARS NO LIABILITY FOR ANY DAMAGES RESULTING FROM USE (OR ATTEMPTED USE AFTER THE EXPIRATION DATE) OF THE SOFTWARE PRODUCT, AND HAS NO DUTY TO PROVIDE ANY SUPPORT BEFORE OR AFTER THE EXPIRATION DATE OF AN EVALUATION LICENSE.

1b. NON-COMMERCIAL USE OF SOFTWARE PRODUCT

„Non-Commercial Use" means the use of the Software Product for non-commercial purposes only, and is limitated to the following users: (a) non-profit organizations (charities and other organizations created for the promotion of social welfare), (b) universities, colleges, and other educational institutions (including, but not limited to elementary schools, middle schools, high schools, and community colleges), (c) independent contractors who are under contract by the above-stated organizations and using the Software Product exclusively for such non-profit or educational clients, and (d) other individual users who use the Software Product for personal, non-commercial use only (for example, hobby, learning, or entertainment).

Under no circumstances can the Software Product be used by or for a for-profit organization, or be used to generate income for personal benefit.

Wingware, a Delaware corporation, reserves the right to further clarify the terms of Non-Commercial Use at its sole determination.

4. GRANT OF NON-EXCLUSIVE LICENSE

Wingware grants the non-exclusive, non-transferable right for a single user to use this Software Product for Non-Commercial Use on a single operating system per software license purchased. Each additional concurrent user of the Software Product, and each additional operating system where the product is used, requires an additional Software Product License.

You may make copies of the Software Product as reasonably necessary for its use. Each copy must reproduce all copyright and other proprietary rights notices on or in the Software Product.

You may install each Software Product License on a single computer system. A second installation of the same Software Product License may be made on one other computer system, so long as both copies of the same Software Product License never come into concurrent use. You may also make copies of the Software Product License as necessary for backup and/or archival purposes. Backup and archival copies may not come into active use, together with the Software Product, for any purpose. No other copies may be made. Each copy must reproduce all copyright and other proprietary rights notices on or in the Software Product License. You may not modify or create derivative copies of the Software Product License.

All rights not expressly granted to you are retained by Wingware.

3. INTELLECTUAL PROPERTY RIGHTS RESERVED BY WINGWARE

The Software Product is owned by Wingware and is protected by United States and

international copyright laws and treaties, as well as other intellectual property laws and treaties. You must not remove or alter any copyright notices on any copies of the Software Product. This Software Product copy is licensed, not sold. You may not use, copy, or distribute the Software Product, except as granted by this EULA, without written authorization from Wingware or its designated agents. Furthermore, this EULA does not grant you any rights in connection with any trademarks or service marks of Wingware. Wingware reserves all intellectual property rights, including copyrights, and trademark rights.

4. NO RIGHT TO TRANSFER

You may not rent, lease, lend, or in any way distribute or transfer any rights in this EULA or the Software Product to third parties without Wingware's written approval, and subject to written agreement by the recipient of the terms of this EULA.

5. INDEMNIFICATION

You hereby agree to indemnify Wingware against and hold harmless Wingware from any claims, lawsuits or other losses that arise out of your breach of any provision of this EULA.

6. THIRD PARTY RIGHTS

Any software provided along with the Software Product that is associated with a separate license agreement is licensed to you under the terms of that license agreement. This license does not apply to those portions of the Software Product. Copies of these third party licenses are included in all copies of the Software Product.

7. SUPPORT SERVICES

Wingware may provide you with support services related to the Software Product. Use of any such support services is governed by Wingware policies and programs described in online documentation and/or other Wingware-provided materials.

As part of these support services, Wingware may make available bug lists, planned feature lists, and other supplemental informational materials. WINGWARE MAKES NO WARRANTY OF ANY KIND FOR THESE MATERIALS AND ASSUMES NO LIABILITY WHATSOEVER FOR DAMAGES RESULTING FROM ANY USE OF THESE MATERIALS. FURTHERMORE, YOU MAY NOT USE ANY MATERIALS PROVIDED IN THIS WAY TO SUPPORT ANY CLAIM MADE AGAINST WINGWARE.

Any supplemental software code or related materials that Wingware provides to you as part of the support services, in periodic updates to the Software Product or otherwise, is to be considered part of the Software Product and is subject to the terms and conditions of this EULA.

With respect to any technical information you provide to Wingware as part of the support services, Wingware may use such information for its business purposes without restriction, including for product support and development. Wingware will not use such technical information in a form that personally identifies you without first obtaining your permission.

9. TERMINATION WITHOUT PREJUDICE TO ANY OTHER RIGHTS

Wingware may terminate this EULA if you fail to comply with any term or condition of this EULA. In such event, you must destroy all copies of the Software Product and Software Product Licenses.

10. U.S. GOVERNMENT USE

If the Software Product is licensed under a U.S. Government contract, you acknowledge that the software and related documentation are „commercial items," as defined in 48 C.F.R 2.01, consisting of „commercial computer software" and „commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1. You also acknowledge that the software is „commercial computer software" as defined in 48 C.F.R. 252.227-7014(a)(1). U.S. Government agencies and entities and others acquiring under a U.S. Government contract shall have only those rights, and shall be subject to all restrictions, set forth in this EULA. Contractor/manufacturer is Wingware, P.O. Box 1937, Brookline MA 02446-0016, USA.

11. EXPORT RESTRICTIONS

You will not download, export, or re-export the Software Product, any part thereof, or any software, tool, process, or service that is the direct product of the Software Product, to any country, person, or entity -- even to foreign units of your own company -- if such a transfer is in violation of U.S. export restrictions.

12. NO WARRANTIES

YOU ACCEPT THE SOFTWARE PRODUCT AND SOFTWARE PRODUCT LICENSE „AS IS," AND WINGWARE AND ITS THIRD PARTY SUPPLIERS AND LICENSORS MAKE NO WARRANTY AS TO ITS USE, PERFORMANCE, OR OTHERWISE. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, WINGWARE AND ITS THIRD PARTY SUPPLIERS AND LICENSORS DISCLAIM ALL

OTHER REPRESENTATIONS, WARRANTIES, AND CONDITIONS,EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE PRODUCT REMAINS WITH YOU.

13. LIMITATION OF LIABILITY

THIS LIMITATION OF LIABILITY IS TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW. IN NO EVENT SHALL WINGWARE OR ITS THIRD PARTY SUPPLIERS AND LICENSORS BE LIABLE FOR ANY COSTS OF SUBSTITUTE PRODUCTS OR SERVICES, OR FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, OR LOSS OF BUSINESS INFORMATION) ARISING OUT OF THIS EULA OR THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT OR THE FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF WINGWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, WINGWARE'S, AND ITS THIRD PARTY SUPPLIERS' AND LICENSORS', ENTIRE LIABILITY ARISING OUT OF THIS EULA SHALL BE LIMITED TO THE LESSER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE PRODUCT OR THE PRODUCT LIST PRICE; PROVIDED, HOWEVER, THAT IF YOU HAVE ENTERED INTO A WINGWARE SUPPORT SERVICES AGREEMENT, WINGWARE'S ENTIRE LIABILITY REGARDING SUPPORT SERVICES SHALL BE GOVERNED BY THE TERMS OF THAT AGREEMENT.

14. HIGH RISK ACTIVITIES

The Software Product is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring failsafe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software Product, or any software, tool, process, or service that was developed using the Software Product, could lead directly to death, personal injury, or severe physical or environmental damage („High Risk Activities“). Accordingly, Wingware and its suppliers and licensors specifically disclaim any express or implied warranty of fitness for High Risk Activities. You agree that Wingware and its suppliers and licensors will not be liable for any claims or damages arising from the use of the Software Product, or any software, tool, process, or service that was developed using the Software Product, in such applications.

15. GOVERNING LAW; ENTIRE AGREEMENT ; DISPUTE RESOLUTION

This EULA is governed by the laws of the Commonwealth of Massachusetts, U.S.A., excluding the application of any conflict of law rules. The United Nations Convention on Contracts for the International Sale of Goods shall not apply.

This EULA is the entire agreement between Wingware and you, and supersedes any other communications or advertising with respect to the Software Product; this EULA may be modified only by written agreement signed by authorized representatives of you and Wingware.

Unless otherwise agreed in writing, all disputes relating to this EULA (excepting any dispute relating to intellectual property rights) shall be subject to final and binding arbitration in the State of Massachusetts, in accordance with the Licensing Agreement Arbitration Rules of the American Arbitration Association, with the losing party paying all costs of arbitration. Arbitration must be by a member of the American Arbitration Association. If any dispute arises under this EULA, the prevailing party shall be reimbursed by the other party for any and all legal fees and costs associated therewith.

16. GENERAL

If any provision of this EULA is held invalid, the remainder of this EULA shall continue in full force and effect.

A waiver by either party of any term or condition of this EULA or any breach thereof, in any one instance, shall not waive such term or condition or any subsequent breach thereof.

17. OUTSIDE THE U.S.

If you are located outside the U.S., then the provisions of this Section shall apply. Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient redigés en langue anglaise. (translation: „The parties confirm that this EULA and all related documentation is and will be in the English language.“) You are responsible for complying with any local laws in your jurisdiction which might impact your right to import, export or use the Software Product, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

18. TRADEMARKS

The following are trademarks or registered trademarks of Wingware:

Wingware, the dancing bird logo, Wing IDE, Wing IDE 101, Wing IDE Personal, Wing IDE Professional, Wing IDE Enterprise, Wing Debugger, and „Take Flight!“.

19. CONTACT INFORMATION

If you have any questions about this EULA, or if you want to contact Wingware for any reason, please direct all correspondence to: Wingware, P.O. Box 1937, Brookline, MA 02446-0016, United States of America or send email to info at wingware.com.

5.2. Open Source Lizenzinformationen

Wing IDE schließt die folgenden Open Source Technologien ein, von denen die meisten **'OSI Certified Open Source'__** Lizenzen unterliegen, außer wenn es in den Fußnoten anders ausgewiesen ist:

- **'atk'__** -- Toolkit für GUI-Zugänglichkeit von Bill.Haneman, Marc.Mulcahy und Padraig.Obriain -- LGPL [1]
- **'docutils'__** -- reStructuredText Markup-Verarbeitung von David Goodger und Mitarbeitern -- Öffentliche Domäne [2]
- **'expat'__** -- XML verarbeitende Bibliothek von dem Thai Open Source Software Center Ltd, Clark Cooper und Mitarbeitern -- MIT Lizenz
- **'fontconfig'__** -- Erkennungsmechanismus und Support für Schriftartkonfiguration von Keith Packard -- MIT Lizenz
- **'freetype'__** -- Bibliothek für hoch-qualitatives Text-Rendering von Werner Lemberg, David Turner und Mitarbeitern -- FreeType Lizenz
- **'glib'__** -- Bibliothek für Objektentwicklungssupport von Hans Breuer, Matthias Clasen, Tor Lillqvist, Tim Janik, Havoc Pennington, Ron Steinke, Owen Taylor, Sebastian Wilhelmi und Mitarbeitern -- LGPL [1]
- **'gtk+'__** -- GUI-Bibliothek für mehrere Betriebssysteme von Jonathan Blandford, Hans Breuer, Matthias Clasen, Tim Janik, Tor Lillqvist, Federico Mena Quintero, Kristian Rietveld, Søren Sandmann, Manish Singh, Owen Taylor und Mitarbeitern -- LGPL [1]
- **'gtk-engines'__** -- GTK Theme Engines von The Rasterman, Owen Taylor, Randy Gordon -- LGPL [1]
- **'gtkscintilla2'__** -- GTK-Wrapper für Scintilla von Dennis J Houy, Sven Herzberg und Mitarbeitern -- LGPL [1]

- **‘GTK Themen‘__** -- Aero von Marcus Petzoldt, LGPL [1]; Aluminum Alloy von **‘Robert Iszaki‘__** (roberTO), AluminumAlloy License [4]; Glider von Link Dupont, LGPL [1]; Glossy P von m5brane, nicht spezifiziert [5]; gnububble von Kyle Davis, nicht spezifiziert [5]; H2O von Eric R. Reitz, nicht spezifiziert [5]; High Contrast, Low Contrast, und Large Print themes von Bill Haneman und T. Liebeck, LGPL [1]; Redmond and Redmond95 von Anonymous, nicht spezifiziert [5]; Smokey-Blue von Jakub ’jimmac’ Steiner und Paul Hendrick, LGPL [1]; Smooth2000 von ajgenius, nicht spezifiziert [5]; SmoothDesert von Ken Joseph, andere [6]; SmoothRetro von Ken Joseph, andere [6]; SmoothSeaIce von ajgenius, nicht spezifiziert [5]
- **‘gtk-wimp‘__** -- GTK-Thema mit nativem Windows Look von Raymond Penners, Evan Martin, Owen Taylor, Arnaud Charlet und Dom Lachowicz -- LGPL [1]
- **‘libiconv‘__** -- Bibliothek für die Umwandlung von Unicode von Bruno Haible -- LGPL [1]
- **‘libpng‘__** -- Bibliothek für PNG-Bildsupport von Glenn Randers-Pehrson, Andreas Eric Dilger, Guy Eric Schalnat und Mitarbeitern -- zlib/libpng Lizenz
- **‘libXft‘__** -- X Windows Schriftart-Rendering von Keith Packard und Mitarbeitern -- MIT Lizenz
- **‘libXrender‘__** -- X Windows Rendering-Extension von Keith Packard und Mitarbeitern -- MIT Lizenz
- **‘pango‘__** -- Bibliothek für Textlayout und -Rendering von Owen Taylor und Mitarbeitern -- LGPL [1]
- **‘parsetools‘__** -- Python Werkzeuge für Parse-Baum-Umwandlung von John Ehresman -- MIT Lizenz
- **‘py2pdf‘__** -- Konvertierungsprogramm von Python Source-Code in PDF-Ausgabe von Dinu Gherman -- MIT Lizenz
- **‘pygtk‘__** -- Python-Bindings für GTK von James Henstridge und Mitarbeitern -- LGPL [1]
- **‘pyscintilla2‘__** -- Python-Bindings für gtkscintilla2 von Roberto Cavada und Mitarbeitern -- LGPL [1]
- **‘python‘__** -- Die Programmiersprache Python von Guido van Rossum, PythonLabs, und Mitarbeitern -- Python 2.3 Lizenz [3]
- **‘render‘__** -- Kopfdateien für X Render-Extension von Keith Packard -- MIT Lizenz

- **'scintilla'__** -- Source-Code-Editorkomponente von Neil Hodgson und Mitarbeitern -- MIT Lizenz
- **'zlib'__** -- Bibliothek für Datenkomprimierung von Jean-loup Gailly und Mark Adler -- zlib/libpng Lizenz

Hinweise

[1] Die LGPL erfordert, dass wir den Source-Code für alle Bibliotheken, die zu Wing IDE verbunden sind, weiterverteilen. Alle diese Module sind im Internet verfügbar. In einigen Fällen können wir Änderungen vorgenommen haben, die noch nicht in die offiziellen Versionen aufgenommen wurden; wenn Sie eine Kopie unserer Version des Source-Codes für irgendeines dieser Module möchten, senden Sie uns bitte eine E-Mail an **'info at wingware.com'__**.

[2] Docutils enthält einige Teile, die anderen Lizenzen unterliegen (BSD, Python 2.1, Python 2.2, Python 2.3 und GPL). Siehe die COPYING.txt Datei in der Source-Code-Verteilung für Einzelheiten.

[3] Die Python 2.3 Lizenz ist eine OSI anerkannte Open Source Lizenz. Jede Version von Python unterliegt einer ähnlichen, aber einzigartigen Lizenz; Wing enthält nur Python 2.3.

[4] Nicht OSI anerkannt. Wingware hat vom Autor die ausdrückliche Erlaubnis erhalten, diese Themen weiterzuverteilen.

[5] Nicht OSI anerkannt. Diese GTK-Themen sind weit verteilte Arbeiten, die impliziet in der öffentlichen Domäne sind, aber keine angegebene Lizenz oder Copyright haben. Sie können von Wing IDE entfernt werden, ohne die grundsätzliche Funktionalität des Produktes durch das Entfernen der entsprechend benannten Verzeichnisse aus bin/gtk-bin/share/themes innerhalb der Wing IDE Installation zu ändern.

[6] Nicht OSI anerkannt. Diese Lizenz umfasst jedoch das Recht, sie ohne Beschränkungen zu ändern und zu verwenden.

Scintilla Copyright

Die Lizenzbedingungen von Scintalla verlangen, dass wir die folgende Copyright-Anmerkung in dieser Dokumentation einschließen:

Fontconfig Copyright

Die Lizenzbedingungen von Fontconfig verlangen, dass wir die folgende Copyright-Anmerkung in dieser Dokumentation einschließen:

Docutils System Messages

Anonymous hyperlink mismatch: 110 references but 112 targets. See „backrefs“ attribute for IDs.